

UNIVERSIDAD AUTÓNOMA CHAPINGO

**POSGRADO EN INGENIERÍA AGRÍCOLA
Y USO INTEGRAL DEL AGUA**

**SIMULACIÓN DE CRECIMIENTO DE TOMATE DE CÁSCARA
(*Physalis ixocarpa*, Brot.) MEDIANTE LOS MODELOS SUCROS Y
LINTUL**

TESIS

**QUE COMO REQUISITO PARCIAL
PARA OBTENER EL GRADO DE:
MAESTRO EN INGENIERÍA AGRÍCOLA Y USO
INTEGRAL DEL AGUA
ORIENTACIÓN: BIOSISTEMAS**



RECCION GENERAL ACADÉMICA
DEPTO. DE SERVICIOS ESCOLARES
CIMA DE EXAMENES PROFESIONALES

**PRESENTA:
CARLOS ROBLES BAÑUELOS**

Chapingo, México, diciembre de 2010

**SIMULACIÓN DE CRECIMIENTO DE TOMATE DE CÁSCARA
(*Physalis ixocarpa*, Brot.) MEDIANTE LOS MODELOS SUCROS Y
LINTUL**

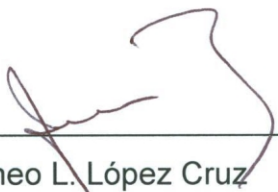
Tesis realizada por Carlos Robles Bañuelos bajo la dirección del comité asesor indicado, aprobada por el mismo y aceptada como requisito parcial para obtener el grado de:

MAESTRO EN INGENIERÍA AGRÍCOLA Y USO

INTEGRAL DEL AGUA

ORIENTACIÓN: BIOSISTEMAS

DIRECTOR:



Dr. Irineo L. López Cruz

ASESOR:



Dr. Ramón Arteaga Ramírez

ASESOR:



Dr. Mario Alberto Vázquez Peña

AGRADECIMIENTOS

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico para la realización de esta investigación mediante una beca para tesis de posgrado.

A la Universidad Autónoma Chapingo (UACH) y al posgrado en Ingeniería Agrícola y Uso Integral del Agua (IAUIA) por darme la oportunidad de realizar los estudios de maestría.

Al Dr. Irineo I. López Cruz por su paciencia, su apoyo y el tiempo dedicado para llevar la dirección del presente trabajo.

Al Dr. Ramón Arteaga Ramírez y el Dr. Mario A. Vázquez Peña por su colaboración y tiempo dedicado para poder llevar a cabo este trabajo.

A la Ing. Maricela Posadas, Ing. Malinalli Achiquen, Ing. Rudy Vázquez, Ing. Omar Arenas y Marco Antonio Robles por su apoyo y colaboración en la realización del trabajo.

A los alumnos del posgrado en ingeniería agrícola y uso integral del agua por su apoyo, amistad y compañerismo.

A aquellas personas que siempre han estado a mi lado, brindándome su apoyo y que hoy contribuyen a consumir este trabajo.

DATOS BIOGRÁFICOS

Nace el 30 de noviembre de 1985 en la comunidad de José Ma. Morelos, Chalchihuites, Zacatecas. Realiza sus estudios de preparatoria agrícola y la licenciatura en la Universidad Autónoma Chapingo, obteniendo el título de Ingeniero en Irrigación en el año 2008.

Sus áreas de investigación son los sistemas de riego agrícola y la modelación en biosistemas.

SIMULACIÓN DE CRECIMIENTO DE TOMATE DE CÁSCARA (*Physalis ixocarpa*, Brot.) MEDIANTE LOS MODELOS SUCROS Y LINTUL

GROWTH SIMULATION OF HUSK TOMATO (*Physalis ixocarpa*, Brot.) USING THE SUCROS AND LINTUL MODELS

C. Robles-Bañuelos¹; I. L. López-Cruz²

El tomate de cáscara es una planta originaria de México con gran importancia nacional, ya que ocupa el quinto lugar en superficie cultivada entre las hortalizas. A pesar de lo anterior, no se ha estudiado si es posible describir su comportamiento mediante modelos matemáticos. Los modelos de simulación del crecimiento de cultivos son de gran utilidad para la investigación agrícola moderna, ya que constituyen una opción importante para la optimización y el control en los sistemas agrícolas. El objetivo de la presente investigación fue la simulación del crecimiento potencial (amplio suministro de agua y nutrientes en un ambiente libre de plagas, enfermedades, contaminantes y malezas) del cultivo de tomate de cáscara (*Physalis ixocarpa*, Brot.) mediante los modelos SUCROS (a Simple and Universal CROp growth Simulator) y LINTUL (Light INterception and UtIlization), así como la simulación de crecimiento alcanzable (crecimiento del cultivo limitado por agua) mediante el modelo SUCROS2, bajo condiciones climáticas de Chapingo, México. El diseño experimental fue completamente al azar con tres tratamientos y tres repeticiones: tres niveles de riego, 60%, 80% y 100% de humedad de acuerdo con el cálculo de la evapotranspiración de referencia (ET_0) con el método de Penman-Monteith. Se hizo un análisis de sensibilidad local para conocer el efecto de los parámetros sobre el comportamiento de los modelos. Se llevaron a cabo calibraciones de los modelos para la variable de salida peso seco total, así como para las variables de estado: peso seco de hojas, tallos, raíces, órganos de almacenamiento (frutos) e índice de área foliar, modificando los valores de un subconjunto de parámetros para el cual los modelos resultaron ser más sensibles. Para la estimación de los parámetros, se usó el algoritmo de mínimos cuadrados (*lsqnonlin.m*) disponible en Matlab para ajustar las predicciones a las mediciones. Los resultados mostraron que los modelos SUCROS, LINTUL y SUCROS2 son capaces de predecir en forma apropiada el crecimiento del tomate de cáscara cultivado en condiciones óptimas y con limitaciones de agua.

Palabras clave: Modelo de crecimiento de cultivos, Crecimiento potencial, crecimiento alcanzable, Modelo matemático, análisis de sensibilidad, calibración.

The husk tomato is a plant native to Mexico with great national importance, as it ranks fifth in cultivated area among vegetables. In spite of this, whether mathematical models can be used to study its behavior has not been studied. Crop growth simulation models are useful for modern agricultural research, as they constitute an important option for optimizing and controlling agricultural systems. The aim of this study was to simulate the potential growth (optimum supply of water and nutrients in an environment free of pests, diseases, contaminants and weeds) of the husk tomato crop (*Physalis ixocarpa*, Brot.) using the SUCROS (a Simple and Universal CROp growth Simulator) and LINTUL (Light INterception and UtIlization) models, as well as its attainable growth (crop growth limited by water) using the SUCROS2 model, under weather conditions of Chapingo, Mexico. A completely randomized experimental design with three treatments and three replicates was used: three irrigation levels, 60%, 80% and 100% humidity according to the calculation of reference evapotranspiration (ET_0) with the Penman-Monteith method. A local sensitivity analysis was carried out in order to know the effect of the model's parameters on the model's behavior. Models were calibrated taking into account the output variable total dry weight, as well as the state variables: dry weight for leaves, stems, roots, storage organs (fruits) and leaf area index, by changing the values of a subset of parameters for which the models were more sensitive. For parameter estimation, ordinary least squares algorithm (*lsqnonlin.m*) available in Matlab was used to obtain a better fit between predictions and measurements. Results showed that SUCROS, LINTUL and SUCROS2 models predict properly the growth of a husk tomato crop grown in optimal conditions and with limited water.

Keywords: Crop growth model, potential growth, attainable growth, mathematical model, sensitivity analysis, calibration.

¹ Tesista

² Director

ÍNDICE GENERAL

Resumen.....	v
Índice de cuadros.....	ix
Índice de figuras.....	xii
Capítulo 1. Introducción.....	1
Capítulo 1. Revisión de literatura.....	4
2.1. Modelos para crecimiento y desarrollo de cultivos anuales.....	4
2.1.1. Objetivos de los modelos matemáticos de cultivos.....	7
2.1.2. Tipos de modelos para simulación de crecimiento de cultivos	9
2.1.3. Modelo para crecimiento potencial de cultivos SUCROS971...	12
2.1.4. Modelo simple para crecimiento potencial de cultivos LINTUL1	14
2.1.5. Modelo para crecimiento limitado por agua SUCROS972.....	16
2.1.6. Otros modelos para simulación de crecimiento de cultivos.....	16
2.2. Aspectos generales sobre el cultivo de tomate de cáscara.....	26
2.2.1. Descripción botánica.....	26
2.2.2. Descripción fisiológica.....	26
2.3. Análisis de sensibilidad y estimación de parámetros en modelos para crecimiento de cultivos.....	27
2.3.1. Incertidumbre y análisis de sensibilidad de un modelo.....	28
2.3.2. Calibración de un modelo dinámico.....	33
2.3.3. Método de Mínimos Cuadrados Ordinarios (OLS).....	35
Capítulo 3. Materiales y métodos.....	39

ÍNDICE GENERAL (Continuación)

3.1. Localización y descripción del campo experimental.....	39
3.2. Descripción del experimento y obtención de datos.....	40
3.2.1. Datos climatológicos necesarios para la modelación.....	43
3.3. Estructura general de los modelos SUCROS971 y LINTUL1.....	44
3.4. Estructura general del modelo SUCROS972.....	51
3.5. Análisis de sensibilidad local.....	55
3.6. Calibración de los modelos SUCROS971, LINTUL1 y SUCROS972	56
Capítulo 4. Resultados y discusión.....	57
4.1. Análisis de sensibilidad para los modelos de crecimiento potencial.....	57
4.2. Calibración automática de los modelos para crecimiento potencial.....	63
4.2.1. Calibración del modelo SUCROS971.....	63
4.2.2. Calibración del modelo simple de crecimiento potencial LINTUL1.....	72
4.3. Análisis de sensibilidad para el modelo de crecimiento limitado por agua SUCROS972.....	80
4.4. Calibración del modelo SUCROS972 para 60% de humedad.....	84
4.5. Calibración del modelo SUCROS972 para 80% de humedad.....	87
Capítulo 5. Conclusiones.....	91
Literatura citada.....	93

ÍNDICE GENERAL (Continuación)

Apéndice A. Simulaciones para las variables de estado del modelo de crecimiento potencial SUCROS971.....	103
Apéndice B. Simulaciones para las variables de estado del modelo de crecimiento potencial LINTUL1.....	109
Apéndice C. Nomenclatura.....	114
Apéndice D. Implementación C-Mex s-function del modelo SUCROS971 para la simulación del crecimiento potencial del cultivo de tomate de cáscara.....	117
Apéndice E. Implementación C-Mex s-function del modelo LINTUL1 para la simulación del crecimiento potencial del cultivo de tomate de cáscara.....	130
Apéndice F. Implementación C-Mex s-function del modelo SUCROS972 para la simulación del crecimiento limitado por agua del cultivo de tomate de cáscara.....	139

ÍNDICE DE CUADROS

3.1.	Fechas y número de muestreos realizados durante el ciclo del cultivo.....	43
4.1.	Índices de sensibilidad de los parámetros del modelo de crecimiento potencial SUCROS971 para la variable peso seco total.....	57
4.2.	Índices de sensibilidad de los parámetros del modelo simple de crecimiento potencial LINTUL1.....	60
4.3.	Calibración automática del modelo SUCROS971 para la variable peso seco total.....	64
4.4.	Calibración automática del modelo SUCROS971 para la variable peso seco de hojas.....	65
4.5.	Calibración automática del modelo SUCROS971 para la variable peso seco de tallos.....	66
4.6.	Calibración automática del modelo SUCROS971 para la variable peso seco de raíces.....	67
4.7.	Calibración automática del modelo SUCROS971 para la variable peso seco de frutos.....	68
4.8.	Calibración automática del modelo SUCROS971 para la variable índice de área foliar.....	69

ÍNDICE DE CUADROS (Continuación)

4.9.	Raíz cuadrada del cuadrado medio del error (RMSE) y coeficiente de determinación (R^2) de la ecuación de regresión entre variables medidas y estimadas obtenidos de la calibración del modelo SUCROS971.....	70
4.10.	Valores de los parámetros del modelo SUCROS971 después de la calibración usando el método de mínimos cuadrados.....	71
4.11.	Calibración automática del modelo LINTUL1 para la variable peso seco total.....	73
4.12.	Calibración automática del modelo LINTUL1 para la variable peso seco de hojas.....	74
4.13.	Calibración automática del modelo LINTUL1 para la variable peso seco de tallos.....	75
4.14.	Calibración automática del modelo LINTUL1 para la variable peso seco de raíces.....	76
4.15.	Calibración automática del modelo LINTUL1 para la variable índice de área foliar.....	77
4.16.	Raíz cuadrada del cuadrado medio del error (RMSE) y coeficiente de determinación (R^2) de la ecuación de regresión entre variables medidas y estimadas obtenidos de la calibración del modelo LINTUL1.....	78

ÍNDICE DE CUADROS (Continuación)

4.17. Valores de los parámetros del modelo LINTUL1 después de una calibración usando el método de mínimos cuadrados.....	80
4.18. Índices de sensibilidad de los parámetros del modelo SUCROS972 para crecimiento limitado por agua.....	81
4.19. Calibración automática del modelo de crecimiento limitado por agua para la variable peso seco total (60% de humedad).....	85
4.20. Calibración automática del modelo de crecimiento limitado por agua para la variable peso seco total (80% de humedad).....	88

ÍNDICE DE FIGURAS

2.1.	Relación entre el rendimiento potencial, alcanzable y real, sus imitaciones y factores que reducen el crecimiento. (Bouman, et al., 1996, Rabbinge, 1993).....	6
2.2.	Genealogía de los modelos de simulación de crecimiento de cultivos de 'La Escuela de de Wit', 1965-2000. (Adaptado de Bouman et al. 1996, Xinyou and van Laar, 2005).....	17
2.3.	Diagrama relacional del modelo de simulación de crecimiento de cultivos GECROS, utilizando simbolos de Forrester (1961) (cajas para las variables de estado; válvulas para variables tasa; círculos tipo elipse para variables intermedias; pequeños círculos cruzados para las variables ambientales; líneas continuas con flecha para el flujo de material; y líneas discontinuas con flechas para el flujo de información). El modelo trata de captar la conservación y balance de energía, agua, asimilaciones de carbono (C) y nitrógeno (N). Las variables ambientales de entrada para el modelo incluyen factores climáticos (por ejemplo, radiación y temperatura) y variables edáficas (disponibilidad de agua y nitrógeno, las cuales pueden ser influenciadas por factores climáticos, tales como la lluvia, y por prácticas de manejo como la irrigación y la fertilización). Los coeficientes genéticos son otro tipo de entradas del modelo, pero no se muestran en el diagrama (Xinyou and van Laar 2005).....	19

ÍNDICE DE FIGURAS (Continuación)

3.1.	Campo experimental del Departamento de Irrigación en la Universidad Autónoma Chapingo.....	40
3.2.	Parcelas (repeticiones) con 10 camas de aproximadamente 30 m de largo y 0.6 m de ancho.....	41
3.3.	Variables climatológicas del sitio. (Estación meteorológica Chapingo).....	44
3.4.	Diagramas relacionales que muestran los componentes principales de los modelos SUCROS (izquierda) y LINTUL (derecha).....	50
3.5.	Diagrama relacional que muestran los principales componentes del modelo SUCROS972 (De acuerdo con van Laar et al., 1997).....	54
4.1.	Simulaciones y función de sensibilidad para RGRL (SUCROS971)...	58
4.2.	Simulaciones y función de sensibilidad para EFF (SUCROS971)...	59
4.3.	Simulaciones y función de sensibilidad para KDF (SUCROS971)...	59
4.4.	Simulaciones y función de sensibilidad para AMX (SUCROS971)...	59
4.5.	Simulaciones y función de sensibilidad para SCP (SUCROS971)...	60
4.6.	Simulaciones y función de sensibilidad para SLA (SUCROS971)...	60
4.7.	Simulaciones y función de sensibilidad para RGRL (LINTUL1).....	61
4.8.	Simulaciones y función de sensibilidad para LUE (LINTUL1).....	62
4.9.	Simulaciones y función de sensibilidad para K (LINTUL1).....	62
4.10.	Simulaciones y función de sensibilidad para SLA (LINTUL1).....	62

ÍNDICE DE FIGURAS (Continuación)

4.11. Curvas de crecimiento, sin calibración, y con calibración para la variable peso seco total (SUCROS971).....	64
4.12. Curvas de crecimiento, sin calibración, y con calibración para la variable peso seco de hojas (SUCROS971).....	65
4.13. Curvas de crecimiento, sin calibración, y con calibración para la variable peso seco de tallos (SUCROS971).....	66
4.14. Curvas de crecimiento, sin calibración, y con calibración para la variable peso seco de raíces (SUCROS971).....	67
4.15. Curvas de crecimiento, sin calibración, y con calibración para la variable peso seco de frutos (SUCROS971).....	68
4.16. Curvas de crecimiento, sin calibración, y con calibración para la variable índice de área foliar (SUCROS971).....	69
4.17. Curvas de crecimiento, sin calibración, y con calibración para la variable peso seco total (LINTUL1).....	73
4.18. Curvas de crecimiento, sin calibración, y con calibración para la variable peso seco de hojas (LINTUL1).....	74
4.19. Curvas de crecimiento, sin calibración, y con calibración para la variable peso seco de tallos (LINTUL1).....	75
4.20. Curvas de crecimiento, sin calibración, y con calibración para la variable peso seco de raíces (LINTUL1).....	76

ÍNDICE DE FIGURAS (Continuación)

4.21. Curvas de crecimiento, sin calibración, y con calibración para la variable índice de área foliar (LINTUL1).....	77
4.22. Simulaciones y función de sensibilidad para RGRL (SUCROS972).....	82
4.23. Simulaciones y función de sensibilidad para AMX (SUCROS972).....	82
4.24. Simulaciones y función de sensibilidad para KDF (SUCROS972).....	83
4.25. Simulaciones y función de sensibilidad para EFF (SUCROS972).....	83
4.26. Simulaciones y función de sensibilidad para SLA (SUCROS972).....	83
4.27. Simulaciones y función de sensibilidad para EES (SUCROS972).....	84
4.28. Simulaciones y función de sensibilidad para SCP (SUCROS972).....	84
4.29. Curvas de crecimiento, sin calibración, y con calibración para la variable peso seco total (60% de humedad).....	85
4.30. Simulaciones para las variables de estado de SUCROS972 correspondientes a 60% de humedad.....	86
4.31. Curvas de crecimiento, sin calibración, y con calibración para la variable peso seco total (80% de humedad).....	88
4.32. Simulaciones para las variables de estado de SUCROS972 correspondientes a 80% de humedad.....	89

Capítulo 1. INTRODUCCIÓN

El tomate de cáscara (*Physalis ixocarpa* Brot.) es una especie hortícola originaria de México, por lo que en nuestro país existe gran variedad genética tanto en su forma silvestre como cultivada, siendo actualmente un cultivo de importancia nacional, ya que ocupa el quinto lugar en superficie cultivada entre las hortalizas (Peña et al., 1998). El tomate de cáscara se cultiva en 27 de los 32 estados de la república mexicana (Peña et al., 2002). SARH (2001) reportó que a nivel nacional se cultivaron alrededor de 50 000 ha, con un rendimiento promedio de 12 t/ha.

En la actualidad debido a las alteraciones ambientales y los problemas climáticos que afectan de manera global al medio ambiente y social, la agricultura se ha visto afectada al convertirse en un mercado no rentable, debido a que no se tiene un control sobre las principales variables climáticas que afectan el sistema agrícola. En el caso del cultivo de tomate de cáscara se ha realizado considerable investigación agronómica (López et al, 2009; Moreno et al, 2002; Peña, 2001; Peña et al, 2002, 2008 y Valtierra et al, 2003), sin embargo no se ha estudiado si es posible describir el crecimiento y desarrollo de este cultivo mediante modelos matemáticos.

La actividad científica es inconcebible sin el desarrollo de modelos matemáticos que permitan sintetizar e incrementar el conocimiento existente

sobre un sistema. La investigación agrícola y en especial la ciencia y arte de la horticultura no son la excepción (López et al., 2005).

Los científicos han desarrollado y usado modelos matemáticos de crecimiento y desarrollo de cultivos; cada año se publica en revistas prestigiadas nuevos modelos matemáticos y un análisis de los mismos (Brisson et al., 2003; Stöckle et al., 2003; Xinyou and van Laar, 2005; Keating et al., 2003; Jones et al., 2003; van Laar et al., 1997 y Escuela de Graduados C. T. de Wit de Ecología de la Producción de la Universidad de Wageningen, Holanda, 1997). En México como en otros países en desarrollo se está poco familiarizado con los modelos matemáticos de los sistemas agrícolas y más aún, prevalecen prejuicios que han limitado no solo el uso de los modelos matemáticos generados en los países desarrollados, sino la generación de modelos propios y su empleo en la actividad académica y práctica cotidiana. (López et al., s/f).

Se han desarrollado varios modelos para cultivos en campo abierto con diferentes objetivos, tanto científicos como prácticos, pero en general, los modelos de crecimiento de cultivos han sido desarrollados para formular y probar hipótesis sobre conceptos fisiológicos y agronómicos de los cultivos, para ayudar en la enseñanza y apoyar la toma de decisiones sobre el manejo del sistema agrícola (van Ittersum et al., 2003; Bouman et al., 1996 y López et al., s/f).

Los objetivos del presente trabajo fueron: 1). Simular el crecimiento potencial del cultivo de tomate de cáscara mediante los modelos SUCROS (A Simple and Universal CROp growth Simulator) y LINTUL (Light INTerception

and Utilization simulator). 2). Simular el crecimiento alcanzable (con limitaciones de agua) del cultivo mediante el modelo SUCROS972. 3). Hacer un análisis de sensibilidad local de los modelos para conocer el efecto de los parámetros sobre el comportamiento de los modelos y 4). Calibrar los modelos para ajustar lo más posible las predicciones a las mediciones.

Capítulo 2. REVISIÓN DE LITERATURA

2.1. Modelos para crecimiento y desarrollo de cultivos anuales

Un modelo es una representación simplificada de un sistema, un sistema es una parte limitada de la realidad que contiene elementos interrelacionados, en nuestro caso, un sistema agrícola (cultivo) (de Wit, 1982). Un biosistema es un sistema agrícola de un cultivo.

Un modelo computacional es una representación de un sistema físico u otro sistema de interés, primero expresado matemáticamente y luego implementado en forma de un programa de cómputo, que puede verse como una función de entradas que, al ser evaluada, produce salidas (Morris, 1991).

Un modelo es definido como una representación matemática de un sistema, y la modelación es el proceso de desarrollo de esa representación. Un modelo de un cultivo es el conjunto de relaciones matemáticas que describen los cambios en las variables de estado como resultado de los diversos procesos que ocurren (Peart and Bruce, 1998).

Un sistema es una parte bien delimitada del mundo real. Por ejemplo un cultivo con todos sus órganos (raíces, tallo, hojas), sus procesos y mecanismos (crecimiento, desarrollo, fotosíntesis, transpiración, etc.) constituye un sistema (Bindi, 2003).

Los sistemas biológicos están formados por la interacción de procesos físicos y químicos. Los sistemas vivos están compuestos de muchos subsistemas y componentes, cada uno con sus propias características y comportamiento, contribuyendo a la forma y funcionamiento de un sistema completo. Estos sistemas son muy complejos, muchos componentes interactúan simultáneamente, y sus interacciones son altamente no lineales o de naturaleza caótica. Estas interacciones y no linealidades deben ser tomadas en cuenta cuando se intenta comprender o predecir el comportamiento del sistema.

La simulación es el proceso de la utilización de un modelo o modelos en forma dinámica para seguir los cambios de un sistema a través del tiempo. Los modelos pueden ser descriptivos y/o matemáticos. Técnicamente los modelos son ecuaciones y técnicas que definen y describen un sistema. La simulación consiste en el cálculo de valores en el tiempo de manera dinámica. La simulación de la dinámica de los sistemas agrícolas e industriales se ha convertido en una parte integral de la ciencia agrícola.

La simulación basada en modelos cuantitativos de los procesos biológicos y sus interacciones pueden ayudar considerablemente a comprender el comportamiento de los sistemas vivos y en las formas de manejo de estos para lograr objetivos específicos (Peart and Bruce, 1998).

De Wit, (1982), Penning de Vries, (1989), propusieron una clasificación de sistemas de producción agrícola en cuatro situaciones de producción: producción potencial, producción limitada por agua, producción limitada por

nitrógeno y producción limitada por nutrientes. Recientemente una nueva clasificación de los sistemas de producción agrícola fue introducida en la Escuela de Graduados C. T. de Wit de Ecología de la Producción de la Universidad de Wageningen, Holanda. El crecimiento potencial, está definido por la concentración de dióxido de carbono (CO_2) atmosférico, radiación solar, temperatura y características del cultivo. El crecimiento alcanzable, está determinado por los factores limitantes de agua y nutrientes, además de los considerados para el crecimiento potencial, y el crecimiento real que está determinado por los factores considerados en el crecimiento potencial y alcanzable y además por el efecto de factores como malezas, plagas, enfermedades y contaminantes. Un diagrama se muestra en la Figura 2.1 (Bouman, et al., 1996):

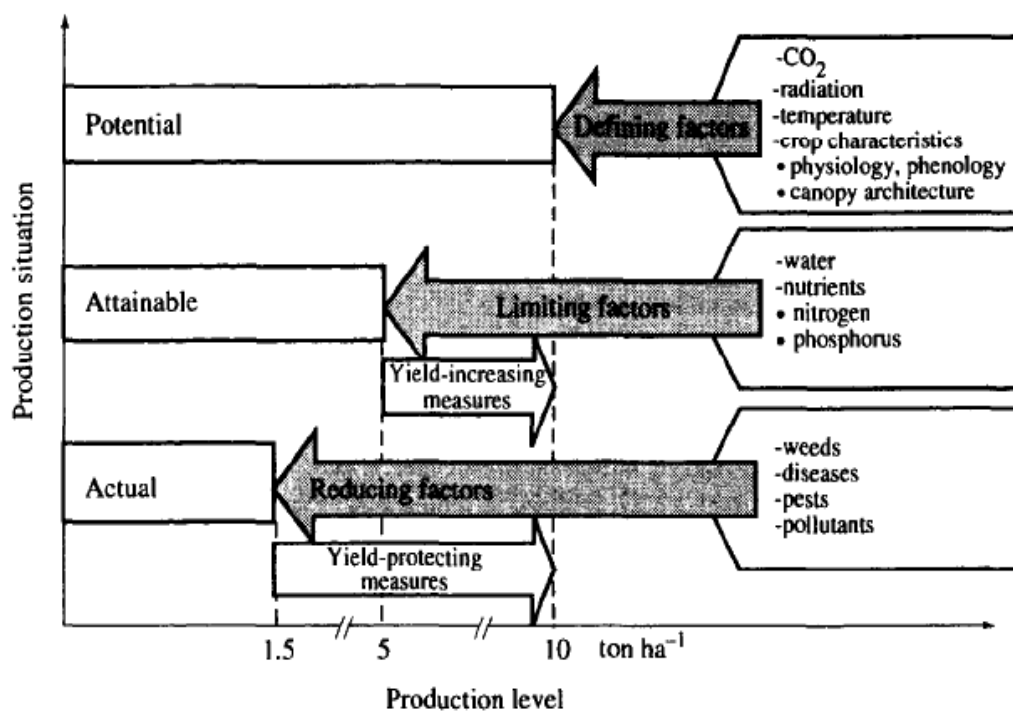


Figura 2.1. Relación entre el rendimiento potencial, alcanzable y real, sus limitaciones y factores que reducen el crecimiento. (Bouman, et al., 1996; Rabbinge, 1993).

En sus principios, la modelación matemática de sistemas agrícolas fue motivada por los cuestionamientos del conocimiento científico y el deseo de cuantificar e integrar los procesos biofísicos para explicar la variación observada en el crecimiento de un cultivo. Posteriormente se hizo énfasis en la investigación agrícola y se han venido adquiriendo conocimientos que han hecho más práctico el uso de modelos de crecimiento de cultivos. El desarrollo de los modelos surgió por la demanda de decisiones tácticas y estratégicas, pronósticos del rendimiento, y estudios de exploración. Algunos de los modelos de crecimiento de cultivos para diferentes situaciones de producción, que se consideran importantes, son por ejemplo: BACROS, SUCROS, WOFOST, MACROS y LINTUL, entre otros. (Bouman et al., 1996).

2.1.1. Objetivos de los modelos matemáticos de cultivos

Los modelos de crecimiento de cultivos han sido desarrollados para formular y probar hipótesis sobre conceptos fisiológicos y agronómicos de los cultivos, para ayudar en la enseñanza y apoyar la toma de decisiones sobre el manejo del sistema agrícola (van Ittersum et al., 2003; Bouman et al., 1996 y López et al., s/f).

En términos generales hay dos objetivos fundamentales para los modelos de simulación biológica:

a). Fines científicos: ampliar el conocimiento del sistema

Los científicos tal vez desean obtener una mejor comprensión del comportamiento de un sistema, las interacciones que ocurren, y las relaciones causa y efecto en un sistema. Por lo tanto, la simulación de sistemas biológicos

puede ayudar a evaluar el entendimiento mediante pruebas de hipótesis acerca de su comportamiento, usando modelos y experimentos bien definidos (Boote et al., 1996; Sinclair, 1990; Monteith, 1990; Goudriaan and Monteith, 1990; Goudriaan, 1988 y Peart and Bruce, 1998).

Un modelo podría servir como una hipótesis científica referente a los procesos de un cultivo y respuestas más generales, y los experimentos serian diseñados específicamente para determinar la adecuación o inadecuación de la hipótesis. Este objetivo fundamental para la modelación podría ser visto como un objetivo científico, ya que el principal resultado esperado es un avance en el conocimiento (Peart and Bruce, 1998).

b). Fines prácticos: optimizar y controlar el sistema

El segundo objetivo está más orientado a problemas de aplicación y podría ser visto como un objetivo de ingeniería. En este caso, el objetivo principal se relaciona con una mejor predicción del comportamiento del sistema, para ser utilizada en el mejoramiento del control y manejo del sistema.

Se puede argumentar que los modelos científicos también se pueden aplicar y que realmente no hay distinción entre estos dos objetivos. Sin embargo, para problemas con orientación y objetivos ingenieriles, es muy importante que el modelo haga una predicción del comportamiento del sistema con un nivel aceptable de precisión.

Los modelos científicos pueden no predecir del todo bien el comportamiento del sistema, pero su propio estudio puede ser muy valioso. Puede demostrar donde el conocimiento es inadecuado y donde se requiere investigación

adicional. La modelación científica es un método altamente eficaz para ayudar a los investigadores a mejorar su comprensión de los sistemas biológicos. El producto final del esfuerzo de la simulación biológica puede ser el incremento en el conocimiento y una mejor comprensión del comportamiento del cultivo, o el producto final puede ser una herramienta que está diseñada para la aplicación a problemas específicos. El reto sería hacer modelos eficaces en brindar información que apoye las decisiones en el sistema (Peart and Bruce, 1998).

Desde la aparición del término *manejo sustentable de la tierra* y el acuerdo general de que muchos sistemas de uso de suelo en todo el mundo son incapaces de garantizar la producción de alimentos para la población mundial cada vez mayor, los modelos de crecimiento de cultivos se han convertido en una herramienta para monitorear los sistemas agrícolas sustentables y para tomar decisiones racionales en cuanto al uso del suelo. (Verdoodt et al, 2004).

2.1.2. Tipos de modelos para simulación de crecimiento de cultivos

Los modelos de simulación del crecimiento de cultivos son instrumentos para reproducir el comportamiento de un cultivo y evaluar su respuesta al ambiente que lo circunda. La construcción de un modelo consiste en la individualización de una serie de ecuaciones matemáticas mediante las cuales es posible reproducir del modo más fiel posible el comportamiento del sistema examinado (Bindi, 2003).

En general, se reconocen los enfoques teórico y empírico como fundamentales para generar modelos de simulación de crecimiento de cultivos (Poluekov y Topaj, 2001 y Thornley y Johnson, 2000).

Los modelos empíricos o de caja negra son descripciones directas de datos y proporcionan relaciones observables entre las variables de un sistema sin mediar alguna explicación de los mecanismos subyacentes, por tanto son necesarios para describir y resumir datos. Los modelos empíricos más conocidos son los modelos de regresión simple y múltiple o modelos no lineales conocidos como ajuste de curvas.

Los modelos teóricos o mecanicistas son modelos explicativos de un sistema. Normalmente, un conjunto de ecuaciones diferenciales ordinarias es definido para describir el comportamiento de las variables de estado del sistema, las variables que representan las propiedades o atributos del sistema considerado. Los modelos mecanicistas son más apropiados para expresar hipótesis matemáticamente y proporcionan así una descripción cuantitativa y una explicación de los procesos más importantes que ocurren en un sistema de cultivo (Thornley y Johnson, 2000).

Con base en la validez y el significado de las ecuaciones matemáticas es posible distinguir los modelos en: empíricos o descriptivos y fenomenológicos o explicativos. Los modelos empíricos describen de un modo simplificado el comportamiento de un cultivo. El desarrollo de un modelo empírico se basa en la individualización, a partir de datos experimentales, de una o más ecuaciones matemáticas para la representación del proceso examinado. Las principales

carencias de este tipo de aproximación son: la limitada validez en ambientes diversos a los originales y el empleo de ecuaciones que a menudo no tienen un significado biológico (Bindi, 2003).

Un modelo fenomenológico describe en modo cuantitativo los mecanismos y procesos que determinan el comportamiento de un cultivo. Para crear un modelo fenomenológico el cultivo es analizado, sus procesos y mecanismos son cuantificados separadamente. El modelo es construido uniendo e integrando los procesos singulares para todo el cultivo. El número de procesos de principal importancia para la simulación del crecimiento de un cultivo dependen del nivel de detalle que deben alcanzar los resultados y de los factores limitantes del crecimiento (carencia hídrica y/o nutritiva).

El elemento que diferencia los modelos fenomenológicos de los empíricos es la naturaleza y el modo con el cual son caracterizadas las ecuaciones que describen los procesos (Bindi, 2003).

Los modelos dinámicos describen el modo en el cual el sistema cambia en el tiempo y por lo tanto es posible seguir la evolución temporal de cada una de las variables del sistema.

Un modelo de un cultivo es un modelo de un sistema dinámico. La forma general de un modelo de un sistema dinámico en tiempo discreto es:

$$\begin{aligned} U_1(t + \Delta t) &= U_1(t) + g_1[U(t), X(t); \theta] \\ &\vdots \\ U_s(t + \Delta t) &= U_s(t) + g_s[U(t), X(t); \theta] \end{aligned}$$

donde t es el tiempo, Δt es algún incremento de tiempo, $U(t)=[U_1(t),...,U_s(t)]^T$ es el vector de variables de estado en el tiempo t , $X(t)$ es el vector de variables explicativas en el tiempo, θ es el vector de parámetros y g es alguna función. Para modelos de cultivos, Δt es frecuentemente un día. Las variables de estado $U(t)$ pueden incluir por ejemplo índice de área foliar (área foliar por unidad de área de suelo), biomasa, profundidad de raíces, contenido de agua en cada una de las capas del suelo, etc. Las variables explicativas $X(t)$ suelen incluir las condiciones iniciales (como la humedad inicial del suelo), características del suelo (como la cantidad máxima de retención de agua), variables climáticas (como las temperaturas máximas y mínimas diarias) y variables de manejo (como cantidades y fechas de riego) (Wallach et al., 2006).

2.1.3. Modelo para crecimiento potencial de cultivos SUCROS971

El modelo SUCROS (A **S**imple and **U**niversal **C**ROp growth **S**imulator) es uno de los modelos para crecimiento y desarrollo para cultivos más conocido a nivel mundial y es el núcleo de muchos modelos de cultivos anuales usados tanto en investigación como en la práctica, así como lo es LINTUL (**L**ight **I**NTerception and **U**tLiZation simulator).

SUCROS es un modelo mecanicista que explica el crecimiento de cultivos sobre las bases de procesos fisiológicos fundamentales, tales como, asimilación de CO_2 y respiración bajo la influencia de las condiciones ambientales. Su principal objetivo no es la predicción del rendimiento, sino el de incrementar la

comprensión del sistema estudiado, mediante la integración del conocimiento disponible en un modelo de simulación (Goudriaan y van Laar, 1994).

SUCROS971 simula el crecimiento potencial de un cultivo, es decir, la acumulación de materia seca bajo un amplio suministro de agua y nutrimentos en un ambiente libre de plagas, enfermedades y malezas, bajo las condiciones de clima prevalecientes. La tasa de acumulación de materia seca es una función de la radiación, temperatura y características del cultivo. La base para el cálculo de la producción de materia seca, es la tasa de asimilación total de CO_2 (fotosíntesis) del follaje. Esta tasa depende de la cantidad de radiación interceptada por el follaje, la cual es una función de la radiación global o entrante y el área foliar del cultivo.

A partir de la radiación absorbida y las características fotosintéticas de las hojas simples, se calcula la tasa de asimilación diaria de CO_2 , parte de los carbohidratos producidos (CH_2O) se usan para mantener la biomasa actual, los carbohidratos restantes se transforman en materia seca estructural (órganos de la planta). En esta conversión, se pierde biomasa en la forma de respiración de crecimiento. La materia seca producida se reparte entre los distintos órganos de la planta, usando factores de división definidos como una función de la etapa de desarrollo fenológico del cultivo. Los pesos secos de los órganos de la planta son obtenidos mediante la integración de sus tasas de crecimiento en el tiempo. (Goudriaan y van Laar, 1994).

El modelo requiere como variables de entrada las propiedades fisiológicas del cultivo y las condiciones del clima del sitio tales como: latitud geográfica, temperaturas mínimas y máximas diarias y radiación global.

2.1.4. Modelo simple para crecimiento potencial de cultivos LINTUL1

LINTUL1 (Light **INT**erception and **Uti**Lization simulator) simula la producción potencial de materia seca de un cultivo como resultado de la intercepción de luz y la utilización de la constante de eficiencia de uso de la luz, es decir, la acumulación de materia seca bajo un suministro amplio de nutrientes y un ambiente libre de plagas, enfermedades y malezas, bajo condiciones de clima prevalecientes. La tasa de acumulación de materia seca está en función de la radiación y las características del cultivo. El modelo hace uso de la observación común de que la tasa de crecimiento de un cultivo en condiciones favorables, es proporcional a la cantidad de luz interceptada (Escuela de Graduados C. T. de Wit de Ecología de la Producción de la Universidad de Wageningen, Holanda, 1997).

La producción de materia seca es, por lo tanto, modelada como el producto de la intercepción de luz y una constante de eficiencia de uso de la luz. La materia seca producida se reparte entre los distintos órganos de la planta, usando factores de división que están definidos en función de la etapa de desarrollo fenológico del cultivo. El peso seco de los órganos de la planta se obtiene mediante integración de sus tasas de crecimiento en el tiempo.

LINTUL1 requiere como entrada, las propiedades fisiológicas del cultivo y las condiciones ambientales actuales del lugar, caracterizadas por la latitud

geográfica, temperaturas máximas y mínimas diarias y la radiación para cada día del ciclo del cultivo.

Diferencias más relevantes entre SUCROS y LINTUL

SUCROS es un modelo mecanicista que permite describir el crecimiento potencial de un cultivo en función de procesos fisiológicos como fotosíntesis y respiración, la tasa de crecimiento depende directamente de la tasa de asimilación de CO₂ del dosel. Su valor diario se calcula a partir de la radiación absorbida por el follaje y las propiedades fotosintéticas de las hojas individuales. Parte de los carbohidratos producidos por la fotosíntesis se usan para mantener la biomasa existente, el resto es empleado para formar nueva biomasa estructural. Además se considera una pérdida de biomasa en la respiración de crecimiento. LINTUL es un modelo similar a SUCROS pero más simplificado, ya que no realiza un cálculo detallado de las tasas de fotosíntesis y respiración, sino que relaciona la cantidad de biomasa en forma proporcional a la cantidad de radiación interceptada por el dosel (López et al, 2008).

El modelo simple LINTUL describe el crecimiento de un cultivo con base en la intercepción y utilización de luz, y la distribución de materia seca con base en un índice de cosecha, mientras que el modelo más detallado SUCROS describe el crecimiento con base en la fotosíntesis y la respiración, y asigna los incrementos de materia seca diaria a los distintos órganos de la planta de acuerdo con factores de división introducidos como una función de la etapa de desarrollo del cultivo (Spitters, 1990). Debido a que el modelo SUCROS es más detallado, es por lo tanto, más complejo, ya que toma en cuenta más procesos

fisiológicos en su estructura. El modelo SUCROS es más teórico, por lo que es más utilizado para fines científicos (enseñanza e investigación), el modelo LINTUL es más utilizado en la práctica, debido a su simplicidad.

2.1.5. Modelo para crecimiento limitado por agua SUCROS972

SUCROS972 describe la producción bajo condiciones limitadas de agua mediante un balance hídrico del cultivo y el suelo. Las condiciones son óptimas con respecto a otros factores de crecimiento, es decir, un cultivo libre de malezas, plagas y enfermedades, así como un amplio suministro de nutrientes. Con el modelo SUCROS972 se pueden estudiar los efectos de escasez de agua temporales, así como las posibilidades de conservación de suelo y agua. SUCROS972 solo puede entenderse a través de SUCROS971 (el modelo de crecimiento de cultivos para producción potencial).

2.1.6. Otros modelos para simulación de crecimiento de cultivos

Resumen breve de los modelos existentes

Desde el trabajo pionero de de Wit (1965), quien introdujo la teoría de sistemas y el enfoque de simulación en sistemas dinámicos en las ciencias agrícolas, un gran número de modelos de simulación de crecimiento de cultivos 'Wageningen' con varios niveles de sofisticación han sido desarrollados (Figura 2.2). Un modelo simple ELCROS (Elementary CROp growth Simulator) fue el primero desarrollado por de Wit et al (1970). Más tarde le siguió un modelo más completo BACROS (Basic CROp growth Simulator), prediciendo la fotosíntesis del dosel, la transpiración y la producción del cultivo con tamaños de paso de una hora o más cortos (Xinyou and van Laar, 2005).

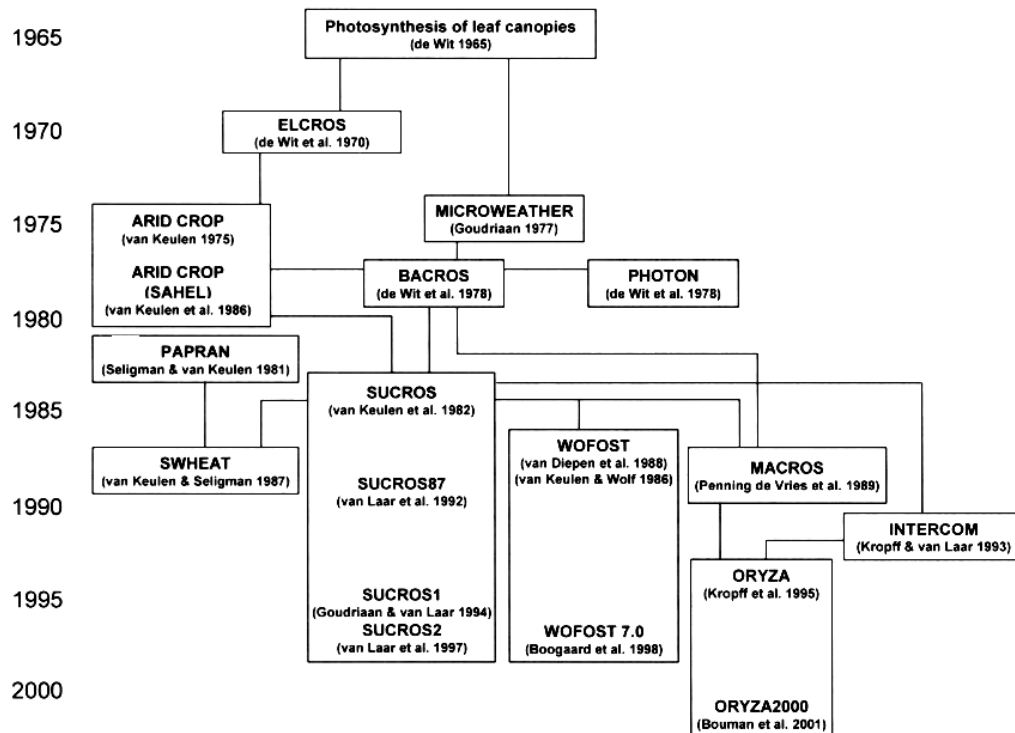


Figura 2.2. Genealogía de los modelos de simulación de crecimiento de cultivos de ‘La Escuela de de Wit’, 1965-2000. (Adaptado de Bouman et al. 1996; Xinyou and van Laar, 2005).

Cada vez se hizo más énfasis en entender aplicaciones más específicas, fueron derivados modelos más simples de simulación de crecimiento de cultivos que pudieron ser usados para muchos diferentes cultivos. Tales modelos fueron distinguidos de acuerdo con tres niveles de producción: potencial, limitado por agua, y limitada por nitrógeno (Penning de Vries et al. 1989). Modelos para producción potencial, tales como SUCROS1 - Simple and Universal CROp growth Simulator (van Keulen et al. 1982; Goudriaan and van Laar 1994), que suponen que los rendimientos del cultivo están determinados por factores climáticos (temperatura y radiación).

Cuando ocurre el estrés hídrico, las tasas de transpiración potencial y real se comparan, y su relación, definida como índice de estrés, es la base para

estimar la fotosíntesis real. Este mecanismo es parte de los modelos de simulación de crecimiento de cultivos para producción limitada por agua, como el modelo SUCROS2 (van Laar et al. 1997).

El modelo de simulación de crecimiento de cultivos GECROS

Es un modelo general (Genotype-by-Environment interaction on CROp growth Simulator), basado en líneas de investigación anteriores, puede ser usado para examinar las respuestas de biomasa y producción de proteínas en cultivos herbáceos, tanto para características ambientales como genotípicas. Los parámetros de entrada para el modelo requieren poco muestreo destructivo.

GECROS usa nuevos algoritmos para resumir el conocimiento actual de procesos fisiológicos individuales y sus interacciones y mecanismos de retroalimentación. Intenta modelar cada subproceso a un nivel consistente de detalle, por lo que ningún proceso es exagerado o requiere demasiados parámetros y del mismo modo ningún proceso es tratado de una manera trivial, a menos que sea inevitable por falta de entendimiento. GECROS también trata de mantener un balance entre un modelo de estructura robusta, alta eficiencia computacional y precisión en las salidas del modelo (Xinyou and van Laar, 2005). Un esquema conceptual del modelo GECROS se muestra en la Figura 2.3.

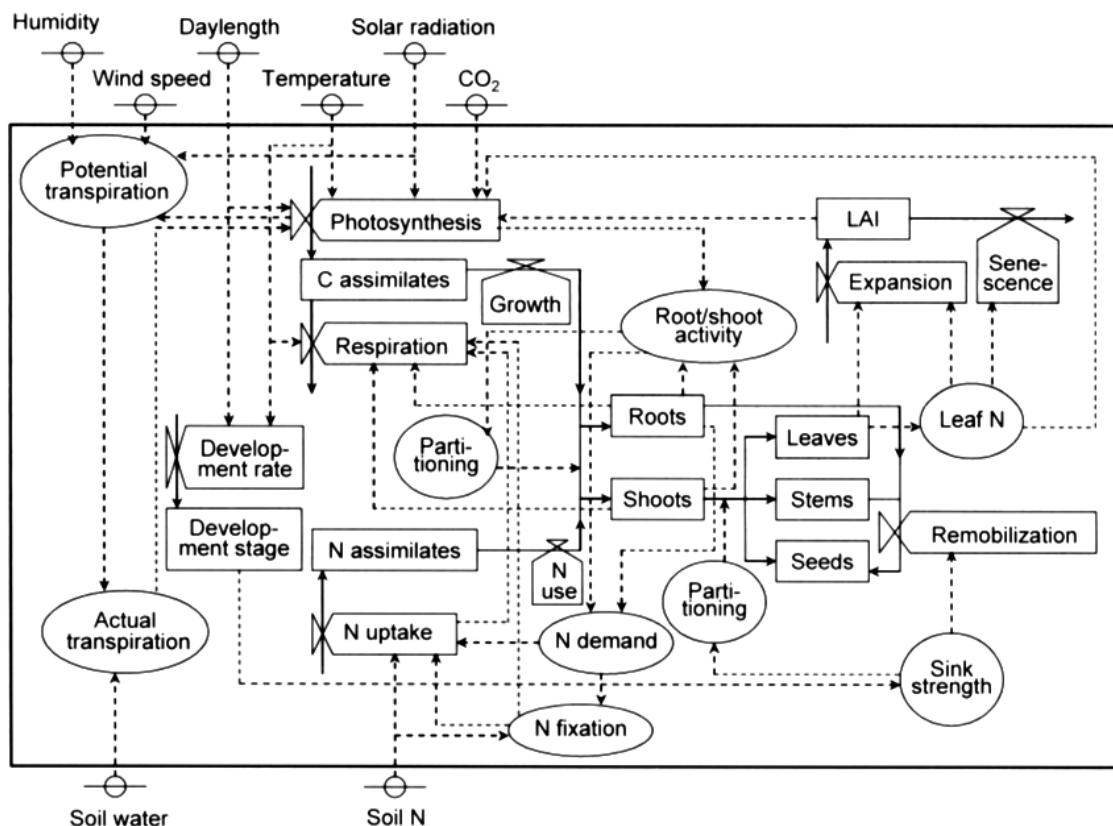


Figura 2.3. Diagrama relacional del modelo de simulación de crecimiento de cultivos GECROS, utilizando símbolos de Forrester (1961) (cajas para las variables de estado; válvulas para variables tasa; círculos tipo elipse para variables intermedias; pequeños círculos cruzados para las variables ambientales; líneas continuas con flecha para el flujo de material; y líneas discontinuas con flechas para el flujo de información). El modelo trata de captar la conservación y balance de energía, agua, asimilaciones de carbono (C) y nitrógeno (N). Las variables ambientales de entrada para el modelo incluyen factores climáticos (por ejemplo, radiación y temperatura) y variables edáficas (disponibilidad de agua y nitrógeno, las cuales pueden ser influenciadas por factores climáticos, tales como la lluvia, y por prácticas de manejo como la irrigación y la fertilización). Los coeficientes genéticos son otro tipo de entradas del modelo, pero no se muestran en el diagrama (Xinyou and van Laar, 2005).

Las características clave de GECROS son:

- Componentes directamente acoplados para el nitrógeno foliar.
- Interacciones carbono-nitrógeno para determinar la partición de raíces y brotes, y las fortalezas para determinar la partición dentro de los brotes.
- Producción de proteína simulada en relación al nitrógeno disponible.

- Aplicable a la mayoría de los cultivos agrícolas y para cualquier situación de producción (ya sea potencial, limitada por agua o limitada por nitrógeno) libre de malezas, plagas y enfermedades.

Modelo de cultivos STICS

STICS (Simulateur mulTIdisciplinaire pour les Cultures Standard) es un modelo que simula el crecimiento de un cultivo, así como los balances de agua y nitrógeno en el suelo mediante datos climáticos diarios. Calcula ambas variables agrícolas (rendimiento, el consumo de entrada) y las variables ambientales (pérdidas de agua y nitrógeno). Desde el punto de vista conceptual, STICS se basa esencialmente en relaciones conocidas o en simplificación de modelos existentes. Uno de los elementos clave de STICS es su adaptabilidad a varios cultivos. Esto se logra mediante el uso de parámetros genéricos apropiados para la mayoría de los cultivos y sobre las opciones en las formalizaciones del modelo acerca de la fisiología y el manejo, éstas deben ser elegidas para cada cultivo (Brisson et al., 2003).

Los datos para ejecutar el modelo se relacionan con el clima, el suelo (perfiles iniciales de agua y nitrógeno y características permanentes del suelo) y el manejo del cultivo. Las especies y variedades de parámetros son proporcionadas por los especialistas de cada especie. STICS es un modelo de cultivos con tamaño de paso diario con variables de entrada en relación al clima, el suelo y el sistema de cultivo. Sus variables de salida están relacionadas con el rendimiento en términos de cantidad y calidad y el ambiente en términos de drenaje y lixiviación de nitratos. El objeto simulado es la

situación del cultivo para la cual un medio físico y un programa de manejo del cultivo pueden ser determinados. Los principales procesos simulados son el crecimiento y desarrollo del cultivo, así como los balances de agua y nitrógeno (Brisson et al., 2003).

Modelo de simulación de sistemas de cultivo CropSyst

CropSyst es un modelo con tamaño de paso diario desarrollado para servir como una herramienta analítica para estudiar el efecto del clima, los suelos y el manejo en la productividad de los sistemas de cultivo y el medio ambiente. Simula los presupuestos de agua y nitrógeno en el suelo, el crecimiento y desarrollo del cultivo, el rendimiento del cultivo, la producción de residuos y la descomposición, la erosión del suelo por el agua y la salinidad. El modelo está diseñado para la simulación de crecimiento de un cultivo en un fragmento de tierra, con suelo uniforme, clima, rotación de cultivos y manejo. El crecimiento está descrito a nivel de la planta completa y los órganos. La integración se realiza con tamaño de paso de un día usando el método de Euler.

Una característica clave de CropSyst es la implementación de un simulador de cultivo genérico que permite la simulación de cultivos anuales y multianuales y la rotación de cultivos a través de un único conjunto de parámetros. Las simulaciones pueden durar fracciones de un año o cientos de años.

Además de los componentes principales CropSyst incluye programas de utilidad para estimar los parámetros hidráulicos del suelo y la radiación solar global, y programas para las comparaciones estadísticas de las estimaciones del modelo y los datos medidos. También incluye una librería de enlace

dinámico que puede ser integrada en otros modelos para calcular la evapotranspiración de referencia del cultivo. Es el núcleo de varios programas, contiene todos los objetos necesarios, los procedimientos y funciones para simular la productividad del cultivo, rotación de cultivos en respuesta al clima, al suelo y al manejo (Stöckle et al., 2003).

El Simulador de Sistemas de Producción Agrícola (APSIM)

APSIM es un marco de modelación modular que ha sido desarrollado por la unidad de investigación de sistemas de producción agrícola en Australia. Fue desarrollado para simular procesos biofísicos en sistemas agrícolas, en general donde hay interés en resultados económicos y ecológicos de prácticas de manejo ante las contingencias climáticas. Cuenta con módulos de manejo que incluyen un amplio rango de cultivos, pastos y arboles, procesos en el suelo, incluyendo balance hídrico (Keating et al., 2003).

APSIM es un marco de modelación que permite que módulos individuales de componentes clave del sistema de producción agrícola (definidos por quien desarrollo el modelo y por el usuario del modelo) sean interconectados. APSIM fue diseñado desde el principio como un simulador de sistemas de cultivo que trató de combinar la estimación del rendimiento preciso en respuesta al manejo con predicciones de consecuencias a largo plazo de las prácticas agrícolas en el recurso suelo (por ejemplo, la dinámica de la materia orgánica en el suelo, erosión, acidificación, etc.).

El marco de modelación APSIM está conformado por:

- Un conjunto de módulos biofísicos que simulan procesos biológicos y físicos en sistemas de producción agrícola.
- Un conjunto de módulos de manejo que permiten al usuario especificar las normas y reglas que caracterizan al escenario que está siendo simulado y que controlan la conducta de la simulación.
- Varios módulos para facilitar la entrada y salida de datos de la simulación.
- Un núcleo de simulación que conduce el proceso de simulación y controla todos los mensajes que pasan entre los módulos independientes.

APSIM se ha utilizado en una amplia gama de aplicaciones, incluido el apoyo en la toma de decisiones en la explotación agricultura, el diseño de sistemas agrícolas para la producción o con el objetivo de un buen manejo de los recursos, evaluación del valor de la predicción del clima, análisis de oferta y demanda en las actividades de los negocios agrícolas y como una guía para las actividades de investigación y educación (Keating et al., 2003).

Modelo de simulación de cultivos DSSAT

El sistema de soporte de decisión para la transferencia de agrotecnología (DSSAT) ha estado en uso durante los últimos 15 años por investigadores de todo el mundo. Este paquete incluye modelos de 16 diferentes cultivos con un software que facilita la evolución y aplicación de los modelos de cultivos para diferentes propósitos. Cuenta con un modulo de suelo, una modulo de plantilla de cultivo que puede simular diferentes cultivos mediante la definición de los

archivos de entrada de las especies, una interface para añadir modelos de cultivos individuales si ellos tienen el mismo diseño e interface, un modulo de clima, y un modulo para hacer frente a la competencia por la luz y el agua entre el suelo, las plantas y la atmosfera (Jones et al., 2003).

El DSSAT fue originalmente desarrollado por una red internacional de científicos, para facilitar la aplicación de modelos de cultivos en un enfoque de sistemas para la investigación agronómica. Su desarrollo inicial fue motivado por la necesidad de integrar conocimientos acerca del suelo, el clima, los cultivos y el manejo para la toma de mejores decisiones sobre la transferencia de tecnología de producción de un lugar a otro donde el suelo y clima fueran diferentes. El DSSAT es una colección de programas independientes que operan juntos, los modelos de simulación de cultivos son su centro. Las bases de datos describen el clima, el suelo, condiciones del experimento y las mediciones, y la información sobre el genotipo para la aplicación de los modelos para diferentes situaciones. El software ayuda a los usuarios a preparar estas bases de datos y comparar los resultados simulados con las observaciones para darles la confianza en los modelos o para determinar si es necesario hacer modificaciones para mejorar la precisión. El DSSAT ayuda a tomar decisiones, reduciendo el tiempo y los recursos humanos necesarios para el análisis de decisiones alternativas complejas (Jones et al., 2003).

Antes del desarrollo del DSSAT, había modelos de cultivos disponibles, pero éstos fueron utilizados sobre todo en los laboratorios donde fueron creados. Algunos de los modelos originalmente implementados en DSSAT fueron: los

modelos CERES para maíz (Jones and Kiniry, 1986) y trigo (Ritchie and Otter, 1985) y la soya SOYGRO (Wilkerson et al., 1983), el cacahuate PNUTGRO (Boote et al., 1986).

Los objetivos del modelo de simulación de cultivos del sistema de soporte de decisión para la transferencia de agrotecnología (DSSAT-CSM) son:

- Simular los sistemas de producción de un monocultivo tomando en cuenta el clima, la genética, el agua en el suelo, el carbono y nitrógeno del suelo, el manejo en las estaciones individuales o múltiples y en la rotación de cultivos en cualquier lugar donde los insumos mínimos sean proporcionados.
- Proporcionar una plataforma para la fácil incorporación de módulos para otros factores abióticos y bióticos, tales como el fósforo en suelo y las enfermedades de las plantas.
- Proporcionar una plataforma que permita comparar fácilmente módulos alternativos para los componentes específicos para facilitar la mejora del modelo, su evolución y documentación.
- Proporcionar la capacidad para introducir fácilmente el modelo de simulación del cultivo (CSM) en los programas de aplicación de una manera modular y bien documentada.

El DSSAT-CSM simula el crecimiento, desarrollo y rendimiento de un cultivo que está creciendo en un área uniforme de suelo bajo manejo prescrito o simulado, así como los cambios en la humedad del suelo, carbono, y nitrógeno que tomarán lugar en el sistema de cultivo en el tiempo (Jones et al., 2003).

2.2. Aspectos generales sobre el cultivo de tomate de cáscara

2.2.1. Descripción Botánica

La especie *Physalis ixocarpa*, es una planta herbácea anual, de 40 a 90 cm de altura, dependiendo del habito de crecimiento, produce ramas y hojas de textura suave con un ciclo de vida de 85 a 90 días. Su raíz es típica o columnar, presenta ramificaciones secundarias profundas que pueden alcanzar hasta 60 cm o más. El tallo es estirado, herbáceo o ligeramente leñoso en la base; ramas primarias de 0.08 a 2.3 cm de diámetro; en los primeros días de vida se presentan pelos esparcidos en el tallo, hojas y ramas las cuales se pierden a medida que van creciendo (Aguilar y Méndez 2004).

2.2.2. Descripción Fisiológica

Las plantas de tomate de cáscara tienen un ciclo de vida de 70 a 110 días desde la siembra hasta la senescencia dependiendo la variedad, una vez que emerge la plántula inicia un crecimiento lento, aproximadamente 1cm por día; posteriormente a los 25 días, el crecimiento se acelera y se estabiliza alrededor de los 55 que es cuando alcanza una altura de 90 cm (en las plantas rastreras aproximadamente 60 cm), la planta sigue creciendo lentamente y puede llegar alcanzar poco mas de 1m (erectas), esto sucede como a los 70 días, posteriormente la planta empieza a envejecer y cae por el peso de los frutos hasta su muerte.

La temperatura óptima promedio que demanda el tomate de cáscara al momento de sembrar es de 20 a 25° C, su crecimiento vegetativo requiere de 22 a 26° C, pero con temperaturas mayores que 32° C puede provocar una

deshidratación del tubo polínico y en consecuencia abortos y frutos mal formados.

2.3. Análisis de sensibilidad y estimación de parámetros en modelos para crecimiento de cultivos

Un gran número de estudios según (Wallach et al., 2006), tratan con la estimación de parámetros mediante regresión, pero los modelos de cultivos tienen una serie de características que hacen que gran parte de este trabajo sea inaplicable. El problema básico es que los modelos de cultivos generalmente tienen muchos parámetros (>10), y muy a menudo tienen más parámetros que número de datos. Los modelos de cultivos se basan en ecuaciones que describen los procesos que intervienen en el crecimiento y desarrollo del cultivo, y generalmente hay información acerca de estos procesos, por lo tanto, el problema de estimación de parámetros para modelos de cultivos no es un simple problema de regresión, el problema es más bien el uso de ambos, datos de campo e información acerca del crecimiento y desarrollo del cultivo para la estimación de los parámetros del modelo.

Hay distintos métodos para la estimación de parámetros de modelos de cultivos, se pueden distinguir dos enfoques, el frecuentista y el Bayesiano (Wallach et al., 2006). El enfoque frecuentista utiliza métodos de estimación para aproximar el verdadero valor de los parámetros mediante el uso de muestras de datos. Hay diferentes tipos de métodos frecuentistas (máxima verosimilitud, mínimos cuadrados). El Bayesiano estima parámetros de dos diferentes tipos de información, una muestra de datos (como el frecuentista), e

información previa acerca de los valores de los parámetros, el resultado de la aplicación de este método es una distribución de probabilidades de valores de parámetros. Los métodos Bayesianos son atractivos cuando los parámetros tienen un significado biológico o físico. En los modelos de cultivos, la información previa sobre los valores de los parámetros puede obtenerse a partir de estudios anteriores llevados a cabo para analizar el crecimiento y desarrollo del cultivo (Wallach et al., 2006).

2.3.1. Incertidumbre y análisis de sensibilidad de un modelo

Un análisis de sensibilidad (AS) es el estudio de cómo la incertidumbre en la salida de un modelo (numérico o de otro tipo) puede ser asignada a diferentes fuentes de incertidumbre en la entrada del modelo. (Saltelli et al., 2000 y Saltelli, 2002).

Los modelos dinámicos de cultivos se utilizan con frecuencia en ecología, agronomía y ciencias del medio ambiente para la simulación del crecimiento y desarrollo de los cultivos y las variables ambientales a un tiempo de paso discreto. Frecuentemente incluyen un gran número de parámetros cuyos valores son inciertos, y a menudo es imposible estimar todos estos parámetros con precisión (Lamboni et al., 2009). La incertidumbre sobre los parámetros es una de las principales fuentes de incertidumbre en las predicciones del modelo. En consecuencia, la estimación de los parámetros inciertos a partir de datos experimentales es un paso importante y el desempeño del modelo depende en gran medida de la precisión con la cual se estiman los parámetros. (Butterbach-

Bahl et al., 2004; Gabrielle et al., 2006a; Lehuger et al., 2009; Makowski et al., 2006a; Wallach et al., 2001 y Lamboni et al., 2009).

Una de las cuestiones de la utilización de los modelos de crecimiento de cultivos es la precisión de la salida estimada. Las posibles fuentes de incertidumbre son la variabilidad en los parámetros del modelo y la incertidumbre relativa a las ecuaciones definidas. (Cooman y Schrevens, 2006).

El uso de análisis de sensibilidad para seleccionar los parámetros a estimar se basa principalmente en la idea intuitiva de que las predicciones son más exactas cuando los parámetros de mayor influencia se estiman con exactitud (Brun et al., 2001; Tremblay and Wallach, 2004 y Lamboni et al., 2009). El análisis de sensibilidad puede ser un elemento importante del proceso de construcción de modelos (Saltelli et al., 2000).

El objetivo de un análisis de sensibilidad es evaluar la importancia relativa que las variables de entrada, las condiciones iniciales de las variables de estado y los parámetros del modelo tienen sobre la evolución en el tiempo de las variables de estado y las salidas (Saltelli et al., 2000 y López et al., 2004). El análisis de sensibilidad también se puede utilizar para determinar cual subconjunto de los factores de entrada representa la mayor parte de la varianza en la salida (resultados), y en qué porcentaje. (Saltelli et al., 2000).

La identificación y representación de la incertidumbre son componentes esenciales reconocidos en la aplicación de un modelo. En el estudio de incertidumbre, se necesita saber que tanta incertidumbre existe en la salida del

modelo (análisis de incertidumbre) y de donde proviene (análisis de sensibilidad).

El análisis de incertidumbre cuantifica la variabilidad de la salida del modelo causada por el conocimiento incompleto o errores del modelista. El análisis de sensibilidad puede ayudar al modelador a medir la capacidad predictiva del modelo (ajuste del modelo), su relevancia, puede ayudar a identificar regiones críticas en el espacio de las entradas, para detectar interacciones entre factores, para establecer prioridades de investigación y para simplificar la estructura del modelo. (Cariboni, 2007).

El análisis de sensibilidad tiene como objetivo establecer la importancia relativa de los factores de entrada que intervienen en el modelo, respondiendo a preguntas tales como:

- ¿Cuál de los factores de entrada en el modelo influyen más en la determinación de la variabilidad que afecta la salida?
- ¿Si la incertidumbre de una de las entradas pudiera ser eliminada, cual debería ser elegida para reducir al mínimo la varianza de los resultados de interés?
- ¿Hay factores cuyo efecto sobre la salida (resultados) es tan bajo, que pueden ser fijados con confianza en cualquiera de los valores dentro de sus rangos de variación, sin afectar los resultados?

Hay dos tipos de análisis de sensibilidad: análisis de sensibilidad local y análisis de sensibilidad global. El análisis de sensibilidad local examina la respuesta local de la salida(s) mediante la variación de los parámetros de

entrada, uno a la vez, manteniendo los otros parámetros en los valores centrales. El análisis de sensibilidad global examina la respuesta global (un promedio de la variación de todos los parámetros) de la salida(s) del modelo mediante la exploración de una región finita (o incluso una infinita). El análisis de sensibilidad local es fácil de implementar. Sin embargo, el análisis de sensibilidad local sólo puede examinar un punto a la vez y el índice de sensibilidad de un parámetro específico depende de los valores nominales (centrales) de los otros parámetros. Por lo tanto, en los estudios más recientes se utilizan más los métodos de análisis de sensibilidad global en lugar del análisis de sensibilidad local. (Chonggang and Zdzislaw, 2007).

Dentro de los métodos numéricos para el cálculo de las sensibilidades locales de un modelo, se encuentra el método de aproximación de diferencias finitas, que es el método utilizado en este trabajo, que es la forma más simple para calcular sensibilidades locales, y está basado en cambiar ligeramente un parámetro a la vez y volver a ejecutar el modelo. Usando este método, los elementos de la matriz de sensibilidad pueden ser aproximados mediante la siguiente ecuación (Saltelli et al., 2000):

$$\frac{\partial y}{\partial k_j} \approx \frac{y(k_j + \Delta k_j) - y(k_j)}{\Delta k_j}, \dots, j = 1, \dots, m.$$

Donde:

- $y(k_j + \Delta k_j)$, Simulación de la variable con el parámetro perturbado
- $y(k_j)$, Simulación de la variable con el parámetro original
- Δk_j , Cambio en el parámetro

Este procedimiento es también llamado el *método de fuerza bruta* o el *método indirecto*. La principal ventaja de este método es que no es necesario modificar el modelo original o extender el código de éste. Sin embargo, el método de la fuerza bruta es más lento y menos preciso que otros métodos más sofisticados.

El cálculo de las sensibilidades locales en esta forma requiere $m + 1$ simulaciones del modelo original. Si se utilizan las diferencias centrales, se requieren $2m$ simulaciones. La precisión de las sensibilidades calculadas depende de la modificación del parámetro Δk_j . En el caso de modelos no lineales, el cambio muy grande en los parámetros (por ejemplo $>5\%$) dañaría el supuesto de linealidad local. Si el cambio en el parámetro es muy pequeño, la diferencia entre las soluciones originales y las perturbadas es muy pequeña y el error de redondeo es muy alto. En la mayoría de los casos, una perturbación del 1% es una buena opción práctica (Saltelli et al., 2000).

Dado que la interpretación de las sensibilidades absolutas es complicada debido a la presencia de unidades, es más conveniente calcular la sensibilidad relativa mediante la ecuación:

$$S_{rel} = \frac{y(k_j + \Delta k_j) - y(k_j)}{\Delta k_j} * \frac{k_j}{y(k_j)}$$

lo cual significa que el parámetro será más importante si éste está más alejado de cero en ambas direcciones.

Debido a que el análisis de sensibilidad local genera funciones de sensibilidad ($S(t)$), para determinar cual parámetro afecta más el comportamiento del modelo se requiere calcular un índice de sensibilidad mediante la siguiente ecuación:

$$I = \int_0^{tf} S_{rel} dt$$

Donde:

I , Índice de sensibilidad de parámetro

S_{rel} , Sensibilidad para la variación de un parámetro j

tf , Tiempo final

2.3.2. Calibración de un modelo dinámico

Los modelos mecanicistas han tomado cada vez más importancia en los últimos años como herramientas de enseñanza e investigación, y como herramientas propuestas para el manejo del cultivo. Estos modelos contienen, en general, un gran número de parámetros (10 o incluso 100 o más).

Frecuentemente hay algo de información inicial acerca de los valores de los parámetros, basada en mediciones específicas para ese parámetro reportado en la literatura. Sin embargo, incluso para los parámetros medibles, se enfrenta a menudo con un rango de valores posibles, debido a un error de medición o debido a diferencias reales para diferentes circunstancias (Wallach et al., 2001).

La acumulación de errores en los diferentes parámetros, además de posibles errores en las ecuaciones de los modelos, pueden conducir a

resultados del modelo que están muy lejos de los datos medidos en campo. Una solución a este problema es calibrar el modelo, que es estimar algunos o todos los parámetros del modelo a partir de datos de campo para mejorar el ajuste entre el modelo y los datos (Wallach et al., 2001).

Estimación de parámetros o calibración de un modelo es el proceso de alterar los parámetros para obtener un mejor ajuste entre datos simulados y medidos (Wallach et al., 2006 y Ruiz, 2009). Un parámetro es un valor numérico que no es calculado por el modelo y que no es una variable de entrada medida u observada (Wallach et al, 2006).

Para los modelos de simulación relativamente complejos, los parámetros se estiman normalmente mediante el ajuste de sus valores hasta que los resultados de simulación y los medidos se ajusten lo más posible. A esto se le conoce como calibración. Muy generalmente, la calibración es un procedimiento para extraer información cuantitativa sobre los parámetros del modelo a partir de observaciones utilizando el modelo de simulación completo (Metselaar, 1999).

La calibración consiste en hacer ajustes a los parámetros del modelo para dar el mejor ajuste entre los resultados simulados y los resultados obtenidos de mediciones en el sistema real. En otras palabras, la calibración implica el ajuste de ciertos parámetros del modelo mediante la comparación sistemática de los resultados simulados con las observaciones de las variables de estado. La estructura del modelo sigue siendo la misma, y los parámetros son ajustados para dar una mejor descripción del comportamiento observado. En muchos

casos, la calibración es únicamente una forma práctica para estimar algunos valores de parámetros que son usados en modelos biológicos (Peart and Bruce, 1998).

El objetivo de realizar la calibración de un modelo es actuar sobre parámetros (o subconjuntos de parámetros) de marcada importancia en la definición de la salida del sistema, modificándolos a fin de obtener un mejor ajuste entre los valores simulados y los observados.

2.3.3. Método de Mínimos Cuadrados Ordinarios (OLS)

Un modelo de cultivo puede ser pensado como una función de respuesta definida por $f(x; \theta)$ donde f es la ecuación del modelo, x es un vector de entradas del modelo y θ es un vector de parámetros del modelo. Esta función nunca predice perfectamente las variables de respuesta. El error esta denotado como $\varepsilon = y - f(x; \theta)$, es la diferencia entre la predicción y la respuesta observada “ y ” para una situación dada. ε es una suma de dos componentes, uno correspondiente al error de medición y otro debido a la insuficiencia de la función de respuesta.

El método de mínimos cuadrados ordinarios es simple y muy comúnmente utilizado por los modeladores de cultivos. Este método tiene buenas propiedades si los errores del modelo están normalmente distribuidos y son independientes con varianza constante y media cero para todos los valores de x . Suponiendo que una muestra de N mediciones, $(x_i; y_i)$ $i=1, \dots, N$, está disponible para la estimación del parámetro del modelo θ . El método más simple para estimar los parámetros del modelo es aplicar (OLS). La estimación

de mínimos cuadrados ordinarios de θ minimiza la suma de diferencias cuadradas entre las mediciones y las predicciones del modelo.

En el caso de modelos lineales es posible calcular analíticamente los cuadrados mínimos ordinarios. La forma general de un modelo lineal es $f(x_i; \theta) = x_i^T \theta$ donde T es la matriz operador transpuesta. Para este tipo de modelo, el estimador OLS de θ es $\hat{\theta}_{OLS} = (X^T X)^{-1} X^T Y$ donde $Y = (y_1, \dots, y_N)^T$ y X es una matriz que incluye los valores medidos de las variables de entrada. Las líneas de X son $x_1^T, \dots, x_i^T, \dots, x_N^T$. $\hat{\theta}_{OLS}$ está relacionando con las observaciones Y a través de una función analítica simple. Esta función puede ser directamente utilizada para calcular los valores de los parámetros estimados a partir de datos.

Esto generalmente es imposible con modelos no lineales; las estimaciones de los parámetros no pueden ser expresadas como funciones de observaciones. El enfoque habitual es utilizar un algoritmo iterativo para minimizar $Z_{OLS}(\theta)$. El siguiente algoritmo (Wallach et al., 2006) es utilizado para resolver este problema:

- (a) Definir un valor inicial para θ , denotado como $\hat{\theta}_0$.
- (b) Linealizar el modelo con la finalidad de sustituir el modelo no lineal por un modelo lineal. En este paso el desarrollo lineal de Taylor se calcula como sigue:

$$f(x; \theta) \approx f\left(x; \hat{\theta}_0\right) + \sum_{j=1}^p \left. \frac{\partial f(x; \theta)}{\partial \theta_j} \right|_{\hat{\theta}_0} \left(\theta_j - \hat{\theta}_0 \right),$$

donde p es el número total de parámetros y $\left. \frac{\partial f(x; \theta)}{\partial \theta_j} \right|_{\hat{\theta}_0}$ es la derivada de

$f(x; \theta)$ con respecto al j -ésimo parámetro tomados en $\hat{\theta}_0$.

(c) Calcular la estimación de los mínimos cuadrados ordinarios de θ para el

modelo lineal $f(x; \hat{\theta}_0) + \sum_{j=1}^p \left. \frac{\partial f(x; \theta)}{\partial \theta_j} \right|_{\hat{\theta}_0} (\theta_j - \hat{\theta}_{0j})$. Esta estimación es obtenida

mediante la minimización de $\sum_{i=1}^N \left\{ y_i - \left[f(x_i; \hat{\theta}_0) + \sum_{j=1}^p \left. \frac{\partial f(x_i; \theta)}{\partial \theta_j} \right|_{\hat{\theta}_0} (\theta_j - \hat{\theta}_{0j}) \right] \right\}^2$.

Ahora como el modelo es lineal, una expresión analítica para el estimador puede ser derivada como se explicó arriba. Se obtiene

$\hat{\theta}_1 = \hat{\theta}_0 + \Delta_0$, donde $\hat{\theta}_1$ es el nuevo parámetro estimado,

$\Delta_0 = (F^T F)^{-1} F^T [Y - F]$, $F = \left[f(x_1; \hat{\theta}_0), \dots, f(x_N; \hat{\theta}_0) \right]^T$, y F es una matriz $(N \times p)$

cuyos elementos están definidos por $\left. \frac{\partial f(x_i; \theta)}{\partial \theta_j} \right|_{\hat{\theta}_0}$.

(d) Sustituir $\hat{\theta}_0$ por $\hat{\theta}_1$ y regresar al inciso a.

Cuando el número de parámetros a estimar es elevado surgen problemas numéricos, en algunos casos el algoritmo no converge a un verdadero valor mínimo de $Z_{OLS}(\theta)$ pero si a un mínimo local. Con algunos valores iniciales, el algoritmo converge a un mínimo local y el valor del parámetro resultante difiere de la estimación de los mínimos cuadrados. Para evitar este tipo de problemas, es importante ejecutar el algoritmo sucesivamente con diferentes valores iniciales.

Otra solución puede ser la utilización de un algoritmo de optimización global, estos algoritmos son menos sensibles a los valores iniciales que el tipo de algoritmo Gauss-Newton y es más probable que converjan a la solución óptima. Una desventaja importante de estos algoritmos es que por lo general requieren de un largo tiempo de cálculo.

Capítulo 3. MATERIALES Y MÉTODOS

3.1. Localización y descripción del campo experimental

Los experimentos de crecimiento del cultivo de tomate de cáscara se realizaron en las instalaciones de la Estación Meteorológica (Figura 3.1) del Departamento de Irrigación en la Universidad Autónoma Chapingo, Chapingo, México, localizada geográficamente a los 19° 29' Latitud norte y 98° 53' Longitud oeste a una altura de 2241 msnm. El clima de la localidad es del tipo Cb(Wo)(W)(i')g, que corresponde a un templado sub-húmedo con lluvias en verano, una época seca en invierno, la temperatura media anual es de 15.5 °C, y la precipitación media anual es de 664 mm. En esta zona el suelo se clasifica como franco-arcilloso, con pH ligeramente alcalino y rico en materia orgánica.

La textura del suelo es franco arcillosa, con densidad aparente de 1.25 y 1.35 g cm⁻³ a las profundidades de 10 y 30 cm respectivamente, la densidad real de 2.35 g cm⁻³ a los 10 cm y 2.39 g cm⁻³ a los 30 cm de profundidad, el porcentaje de humedad a capacidad de campo de 29.6 % y el punto de marchitez permanente de 16.5 %. El espacio poroso o contenido volumétrico de humedad a saturación varía de 0.47 cm³ cm⁻³ a los 10 cm de profundidad y 0.43 cm³ cm⁻³ a los 30 cm (López, 2008).



Figura 3.1. Campo experimental del Departamento de Irrigación en la Universidad Autónoma Chapingo.

3.2. Descripción del experimento y obtención de datos

El experimento se realizó durante el ciclo Primavera-Verano durante los meses de mayo a agosto del año 2009. Para su establecimiento se utilizó planta de tomate de cáscara variedad Diamante, que presenta las siguientes características: la planta es de crecimiento rastrero semirrecto, muy precoz, frutos de tamaño y forma uniforme de color verde alimonado. La siembra se realizó el 18 abril de 2009 en charolas de poli-estireno, el trasplante se llevó a cabo 29 días después el 17 de mayo.

Se utilizó una acolchadora mecánica para hacer camas de siembra de 0.6 m de ancho a 1.5 m de distancia, se colocó cinta de riego (Plastro hydrodrip), con diámetro interno de 15.2 mm, calibre 0.889 mm, 0.3 m de separación entre emisores, gasto del emisor 1.8 L/h. Las camas se cubrieron con acolchado plástico con las siguientes características: color plateado, ancho de 1.2 m,

calibre de 2.28 mm, perforaciones con diámetro de 2.5 pulgadas a 0.45 m de espaciamiento. En la Figura 3.2 se muestran las parcelas con un arreglo de plantación de 1.5 m entre hileras y 0.45 m entre plantas, obteniéndose una densidad de siembra de 14 815 plantas/ha.



Figura 3.2. Parcelas (repeticiones) con 10 camas de aproximadamente 30 m de largo y 0.6 m de ancho.

El diseño experimental fue un completamente al azar con 3 tratamientos (niveles de humedad), con 3 repeticiones, con la finalidad de simular el crecimiento potencial (sin limitaciones de agua, 100% Penman Monteith) y el crecimiento alcanzable con limitaciones de agua (60 y 80% Penman Monteith).

Se aplicaron riegos diariamente (a excepción de días lluviosos) reponiendo la lámina de riego consumida del día anterior, ésta se calculó mediante la ecuación de evapotranspiración de referencia (ET_0) de Penman-Monteith (Allen et al., 1998) basada en la siguiente relación:

$$ET_0 = \frac{0.408 \Delta (R_n - G) + \gamma \frac{900}{T + 273} u_2 (e_s - e_a)}{\Delta + \gamma (1 + 0.34 u_2)}$$

donde R_n es la radiación neta en la superficie del cultivo ($\text{MJ m}^{-2} \text{ d}^{-1}$); G es la densidad de flujo de calor del suelo ($\text{MJ m}^{-2} \text{ d}^{-1}$); T es la temperatura media del aire diario ($^{\circ}\text{C}$); u_2 es la velocidad del viento a 2 m de altura (m s^{-1}); e_s es la presión de vapor a saturación (KPa); e_a es la presión de vapor actual (KPa); Δ es la pendiente de la curva de presión de vapor ($\text{kPa } ^{\circ}\text{C}^{-1}$) y γ es la constante psicrométrica ($\text{kPa } ^{\circ}\text{C}^{-1}$).

A partir del trasplante, cada siete días, durante todo el ciclo y hasta el inicio de senescencia del cultivo se llevaron a cabo muestreos aleatorios destructivos de plantas de tomate con la finalidad de obtener datos de biomasa, así como los valores del índice de área foliar obtenidos mediante un integrador de área foliar MODEL LI-3100; la biomasa se dividió en biomasa de raíz, tallos, hojas y órganos de almacenamiento. Los datos de biomasa a excepción del primer muestreo (muestro 0) fueron requeridos como variables de entrada para la simulación de crecimiento del cultivo mediante los modelos, los datos del primer muestreo fueron tomados como condiciones iniciales.

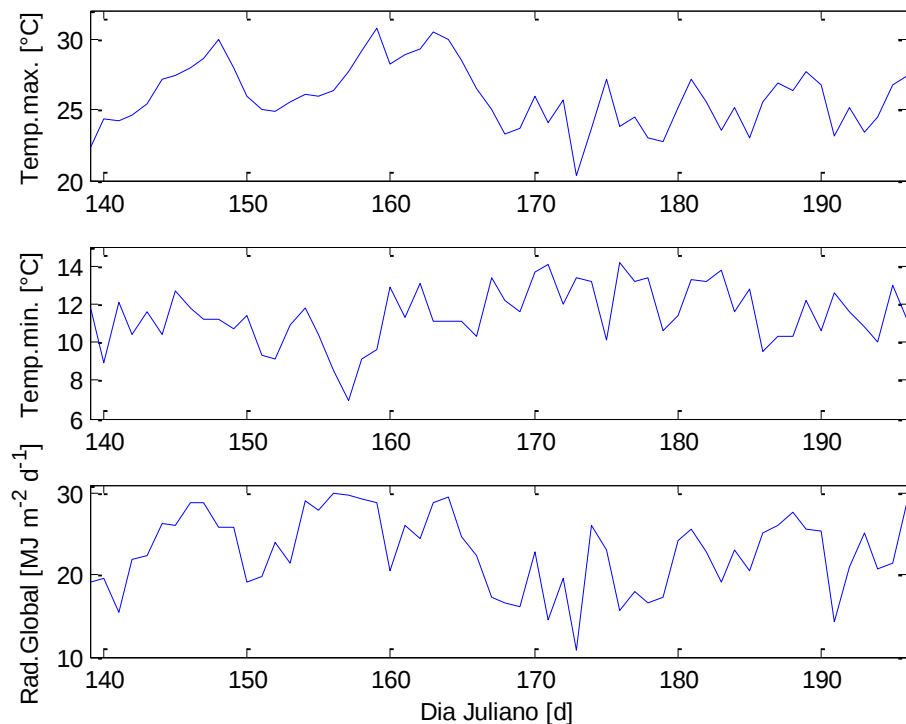
Con la finalidad de que los tratamientos tuvieran condiciones de humedad similares, durante la primera semana después del trasplante se aplicó la misma lámina de riego para los tres tratamientos. En el cuadro 3.1 se muestran las fechas de muestreo.

Cuadro 3.1. Fechas y número de muestreos realizados durante el ciclo del cultivo.

Día juliano	No. muestreo	Fecha	Días después del trasplante
139	0	19 mayo	2
149	1	29 mayo	12
156	2	05 junio	19
163	3	12 junio	26
177	4	26 junio	40
188	5	07 julio	51
196	6	15 julio	59

3.2.1. Datos climatológicos necesarios para la modelación

Los modelos SUCROS y LINTUL requieren como entradas las propiedades fisiológicas del cultivo y las actuales condiciones de tiempo atmosférico en el sitio, caracterizado por su latitud geográfica, temperatura mínima y máxima diaria, radiación, velocidad del viento, presión de vapor, y precipitación para cada día del ciclo del cultivo, las cuales se obtuvieron de la estación meteorológica del sitio (Figura 3.3).



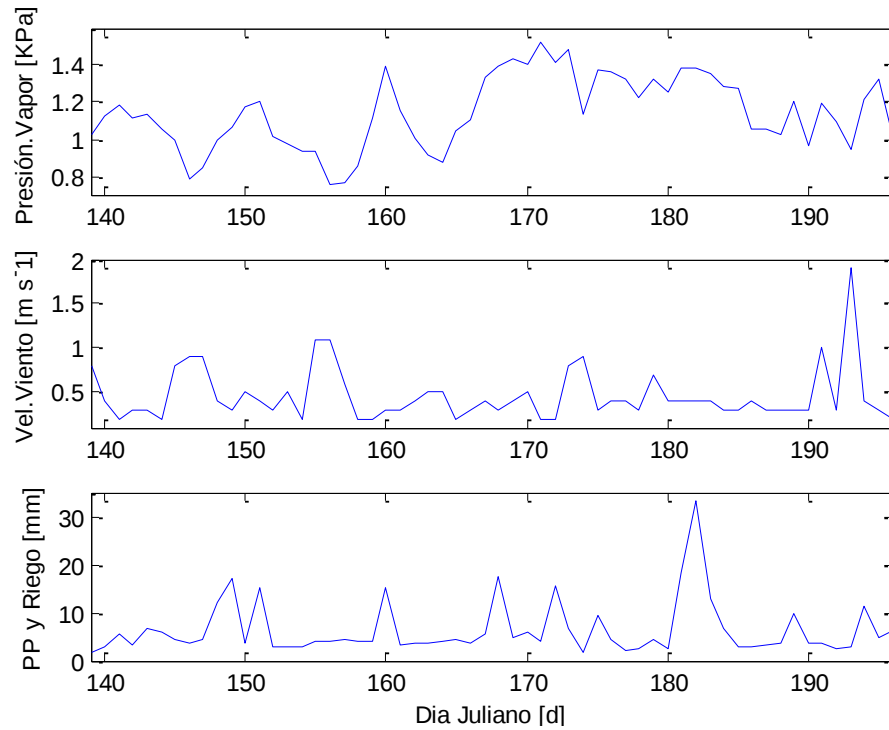


Figura 3.3. Variables climatológicas del sitio. (Estación meteorológica Chapingo).

3.3. Estructura general de los modelos SUCROS971 y LINTUL1

Un modelo de crecimiento y desarrollo para un cultivo en campo puede definirse mediante la ecuación diferencial ordinaria de primer orden no lineal de la forma:

$$\frac{dx}{dt} = f(x, u, p), x(t_0) = \beta$$

Donde: $x \in R^n$, $u \in R^m$, $p \in R^q$ y $\beta \in R^n$. El vector de estados (x) contiene las variables que caracterizan el cultivo como peso seco, número de frutos, área foliar, etc. El vector de las entradas (u) contiene variables tales como temperatura, CO_2 y radiación fotosintéticamente activa. El vector (p) representa los parámetros o constantes fisiológicas y físicas del modelo. Dado que en general f es un vector de funciones no lineales, el modelo de

crecimiento no siempre tiene solución analítica y en esos casos debe ser resuelto mediante integración numérica o simulación digital. A partir de las ecuaciones de estado (1) se pueden calcular otras variables llamadas salidas del sistema (y), la ecuación algebraica general no lineal es entonces:

$$y = g(x, u, p)$$

Algunas variables de salida en cultivos son: peso seco de hojas, el índice de cosecha o la biomasa total de la planta (incluyendo raíces), etc.

En ambos modelos las variables de estado son: el índice de área foliar (x_{iaf} m² m⁻²), la biomasa de raíces (x_{raiz} , g m⁻²), de hojas verdes (x_{hv} , g m⁻²), de hojas secas (x_{hs} , g m⁻²), de tallos (x_{tallo} , g m⁻²) y de los frutos o granos (x_{frutos} , g m⁻²). El estado de desarrollo (x_{edes}) se representa mediante una variable adimensional, donde un valor de cero se asocia a la emergencia, uno para la floración y dos para la maduración en el caso de SUCROS (Goudriaan y van Laar, 1994; van Laar et al., 1997). La ecuación diferencial es:

$$\frac{dx_{edes}}{dt} = f(x_{edes}, T)$$

donde $f(x_{edes}, T)$ es una función empírica de la tasa de desarrollo, que depende de la temperatura media diaria (°C). Sin embargo, en LINTUL en vez del estado de desarrollo, se modela la integral de la temperatura (Escuela de Graduados C. T. de Wit de Ecología de la Producción de la Universidad de Wageningen, Holanda, 1997).

$$\int \max (0, 0.05 (T_{\max} - T_{\min}) - T_{base}) dt$$

La tasa de cambio del índice de área foliar se calcula mediante la ecuación diferencial:

$$\frac{dx_{iaf}}{dt} = Crec_{iaf} - Mort_{iaf}$$

donde la función de la tasa de crecimiento del índice de área foliar ($m^2 m^{-2} día^{-1}$) es:

$$Crec_{iaf} = SLA \cdot Crec_{foliar}$$

si la planta se encuentra en estado de maduración y:

$$Crec_{iaf} = x_{iaf} \cdot (\exp(RGRL \cdot TEFF \cdot \Delta T) - 1) / \Delta T$$

durante la fase de crecimiento juvenil. SLA es el área foliar específica. RGRL es la tasa de crecimiento relativa durante la fase de crecimiento exponencial del área foliar y TEFF es la temperatura efectiva. La tasa de crecimiento de biomasa foliar ($Crec_{foliar}$, $g m^{-2} día^{-1}$) es una función de la tasa de crecimiento de la biomasa del cultivo ($CrecTot$, $g m^{-2} día^{-1}$) afectada por dos funciones que especifican la fracción de biomasa para área foliar ($fr_h(x_{edes})$) y para brotes ($fr_b(x_{edes})$):

$$Crec_{foliar} = fr_h(x_{edes}) \cdot fr_b(x_{edes}) \cdot CrecTot$$

La función de crecimiento de la biomasa total incluyendo traslocación se calcula con la ecuación:

$$CrecTot = (AsimTot - RMant + 0.947 \cdot Transl \cdot CFST \cdot 30/12) / ASRQ$$

donde CFST es la fracción de carbono para los tallos y ASRQ es el requerimiento de asimilados (CH_2O) para la producción de biomasa estructural. La tasa de asimilación bruta diaria ($AsimTot$, $\text{g m}^{-2} \text{ día}^{-1}$) se calcula mediante la ecuación:

$$AsimTot = 30/44 \cdot A_{\max} (1 - \exp(-\varepsilon I / A_{\max}))$$

donde ε es la eficiencia inicial de uso de la luz por las hojas individuales, I es la radiación global diaria, y $A_{\max}$ es la tasa máxima de asimilación de CO_2 . Este cálculo se lleva a cabo mediante una integración Gaussiana en cinco puntos durante el día, con base en una estimación en tres puntos del follaje de la cantidad de radiación absorbida, calculando la radiación fotosintéticamente activa (I , W m^{-2}), tanto difusa como directa a partir de la longitud astronómica del día, la radiación global diaria y el coeficiente de transmisividad atmosférica (Goudriaan y van Laar, 1994). La tasa de respiración de mantenimiento ($RMant$, $\text{g m}^{-2} \text{ día}^{-1}$) se calcula tomando en cuenta la respiración de hojas verdes, tallos, raíces y frutos, es afectada por dos funciones que consideran el efecto de la temperatura del aire y el efecto del estado de desarrollo sobre la tasa de respiración.

$$RMant = (M_{hv} \cdot x_{hv} + M_{tallo} \cdot x_{tallo} + M_{raiz} \cdot x_{raiz} + M_{org} \cdot x_{org}) \cdot f_{resp}(T) \cdot f_{resp}(x_{edes})$$

donde la función $f_{resp}(T)$ da cuenta del efecto de la temperatura sobre la respiración de mantenimiento, $f_{resp}(x_{edes})$ del estado de desarrollo y $f(d_{emerg})$ es igual a la unidad a partir de la emergencia y cero para días anteriores:

$$f_{resp}(T) = Q10^{(T_{avgd} - T_{ref})/10}$$

$$f_{resp}(x_{edes}) = \frac{x_{hv}}{x_{hv} + x_{hs}}$$

La tasa de mortalidad del índice de área foliar ($Mort_{iaf}$, $m^2 m^{-2} día^{-1}$) depende de una tasa relativa de mortalidad (RDR, $día^{-1}$), específica de la especie, en función del estado de desarrollo y el autosombreo del índice de área foliar:

$$Mort_{iaf} = \max(RDRDV, RDRSH) \cdot x_{iaf}$$

donde RDRDV es la razón de muerte relativa debido al desarrollo del envejecimiento y RDRSH es la razón de muerte relativa debido al autosombreo en alta LAI.

Una diferencia fundamental de los modelos SUCROS y LINTUL es el cálculo de la tasa de crecimiento de la biomasa ($CrecTot$, $g m^{-2} día^{-1}$). En LINTUL se supone que el crecimiento es proporcional a la cantidad de radiación solar interceptada por el follaje.

$$CrecTot = LUE \times PAR_{int}$$

donde PAR_{int} ($MJ m^{-2} día^{-1}$) es la cantidad de radiación global interceptada por el dosel y LUE es un parámetro que refleja la eficiencia del uso de la luz (materia seca producida por unidad de radiación fotosintéticamente activa interceptada).

La tasa de cambio de la biomasa de hojas verdes es calculada mediante la ecuación diferencial:

$$\frac{dx_{hv}}{dt} = Crec_{foliar} - Mort_{foliar}$$

donde:

$$Mort_{foliar} = x_{hv} Mort_{iaf} / x_{iaf}$$

La tasa de cambio de la biomasa de hojas muertas se calcula mediante la ecuación:

$$\frac{dx_{hs}}{dt} = Mort_{foliar}$$

La tasa de cambio de la biomasa de raíces se calcula mediante la ecuación diferencial:

$$\frac{dx_{raiz}}{dt} = (1 - fr_b(x_{edes})) \cdot CrecTot$$

La tasa de cambio de la biomasa para tallos se calcula mediante la ecuación:

$$\frac{dx_{tallo}}{dt} = fr_t(x_{edes}) \cdot fr_b(x_{edes}) \cdot CrecTot$$

donde la fracción de la biomasa total asignada a los tallos $fr_t(x_{edes})$ es una función empírica que depende del estado de desarrollo. Finalmente la tasa de cambio para la biomasa de los órganos de almacenamiento (frutos o granos) se calcula mediante la ecuación diferencial:

$$\frac{dx_{frutos}}{dt} = fr_f(x_{edes}) \cdot fr_b(x_{edes}) \cdot CrecTot$$

La traslocación es igual a cero hasta la floración, a partir de entonces se calcula mediante la ecuación:

$$Transl = 0.2 \cdot x_{tallos} f(x_{edes})$$

Las variables de salida en ambos modelos son la biomasa de hojas (y_h , g m⁻²) calculada mediante la ecuación $y_h = x_{hv} + x_{hs}$, la biomasa total de la parte aérea (y_{bas} , g m⁻²), calculada mediante la ecuación algebraica: $y_{bas} = x_{hv} + x_{hs} + x_{tallo} + x_{org}$, la biomasa total de la planta considerando raíces (y_b , g m⁻²), calculada mediante la ecuación algebraica: $y_b = x_{hv} + x_{hs} + x_{tallo} + x_{org} + x_{raiz}$ y el índice de cosecha mediante la ecuación: $y_{hi} = x_{org} / (x_{hv} + x_{hs} + x_{tallo} + x_{org})$. En la Figura 3.4 se muestra un esquema de las variables de estado y los principales componentes de los modelos SUCROS y LINTUL respectivamente; al igual que el modelo SUCROS, LINTUL tiene las mismas variables de estado, sin embargo este último no cuenta con una subrutina tan completa para el cálculo de la fotosíntesis, ya que es un modelo simplificado y muy general que utiliza una constante de eficiencia de uso de la luz.

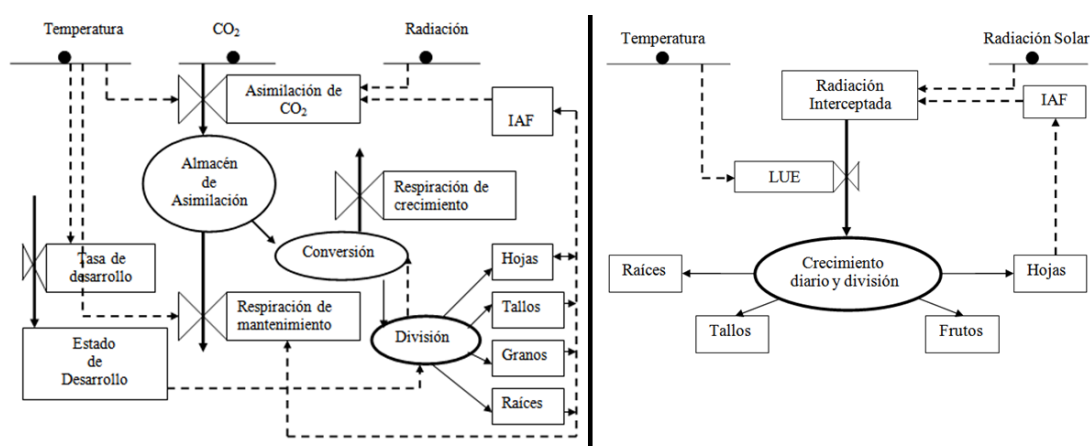


Figura 3.4. Diagramas relacionales que muestran los componentes principales de los modelos SUCROS (izquierda) y LINTUL (derecha).

3.4. Estructura general del modelo SUCROS972

Las más importantes variables de estado en SUCROS2 (van Laar et al., 1997) se describen a continuación. La evapotranspiración potencial total acumulada (x_{etppot} , mm) se calcula mediante la ecuación:

$$\frac{dx_{etppot}}{dt} = TasaEvapAire + TasaEvapRad$$

donde $TasaEvapAire$ (mm d^{-1}) es la evaporación potencial del suelo debida al poder de secado del aire y $TasaEvapRad$ (mm d^{-1}) es la evaporación potencial del suelo debida a la radiación. La conocida ecuación de Penman-Monteith es empleada para calcular la tasa de transpiración potencial. La evaporación potencial acumulada del suelo debida al efecto de la radiación ($x_{tevaprad}$, mm) es calculada mediante:

$$\frac{dx_{tevaprad}}{dt} = TasaEvapRad$$

La evaporación potencial acumulada del suelo debida al efecto de secado del aire ($x_{tevapaire}$, mm) se estima mediante la ecuación diferencial:

$$\frac{dx_{tevapaire}}{dt} = TasaEvapAire$$

La tasa de cambio de la cantidad de agua para cuatro compartimientos de suelo ($x_{w(i)}$, mm) es calculada mediante las ecuaciones siguientes:

$$\frac{dx_{w(1)}}{dt} = \text{Infiltración} - f_{redist}(x_{w(1)}, x_{w(2)}, w(1), w(2)) - EvapRate_{w(1)} - TranspRate_{w(1)}$$

donde la tasa de infiltración (mm d^{-1}) está dada por:

$$\text{Infiltración} = LLuvia - \text{Intercepción} - \text{Esgurrimiento}$$

f_{redist} (mm d⁻¹) es la tasa de redistribución de agua en el suelo calculada mediante el enfoque del 'cubo deposito', $EvapRate_{w(i)}$ (mm d⁻¹) es la tasa de evaporación y $TranspRate_{w(i)}$.

$$\frac{dx_{w(2)}}{dt} = f_{redist}(x_{w(1)}, x_{w(2)}, w(1), w(2)) - f_{redist}(x_{w(2)}, x_{w(3)}, w(2), w(3)) - EvapRate_{w(2)} - TranspRate_{w(2)}$$

$$\frac{dx_{w(3)}}{dt} = f_{redist}(x_{w(2)}, x_{w(3)}, w(2), w(3)) - f_{redist}(x_{w(3)}, x_{w(4)}, w(3), w(4)) - EvapRate_{w(3)} - TranspRate_{w(3)}$$

$$\frac{dx_{w(4)}}{dt} = f_{redist}(x_{w(3)}, x_{w(4)}, w(3), w(4)) - f_{redist}(x_{w(4)}, w(4), MaxDrenaje) - EvapRate_{w(4)} - TranspRate_{w(4)}$$

El drenaje total ($x_{drenaje}$, mm) es calculado mediante la ecuación diferencial:

$$\frac{dx_{drenaje}}{dt} = f_{redist}(x_{w(4)}, w(4), MaxDrenaje)$$

La cantidad de lluvia interceptada por el follaje del cultivo (x_{intpp} , mm) es calculada mediante la ecuación:

$$\frac{dx_{intercepcion}}{dt} = Intercepción$$

donde:

$$Intercepción = \min(Lluvia, 0.25 \times x_{IAF})$$

La cantidad de total de escurrimiento ($x_{escurrimiento}$, mm) se calcula mediante:

$$\frac{dx_{escurrimiento}}{dt} = \max(0, 0.15(Lluvia, Intercepción - 10), Lluvia - Intercepción - f(w(1), x_{w(1)}))$$

La profundidad de raíces (x_{PRaiz} , mm) es calculada mediante una ecuación que depende de varios estados, de la profundidad de las capas del suelo y de la temperatura promedio:

$$\frac{dx_{PRaiz}}{dt} = f(x_{PRaiz}, x_{PRaizMax}, x_{edes}, w(i), T_{media})$$

La evaporación real acumulada del suelo ($x_{EvapReal}$, mm) es estimado mediante la ecuación diferencial:

$$\frac{dx_{EvapReal}}{dt} = TasaEvapReal$$

donde:

$$TasaEvapReal = (1 - \exp(-0.5x_{IAF}))TasaEvapRad + TasaEvapAire \times \min(2.0, x_{IAF}) - 0.5Intercepcion$$

La cantidad total de agua transpirada por el cultivo ($x_{ttransp}$, mm) es calculada mediante la ecuación diferencial:

$$\frac{dx_{ttransp}}{dt} = TranspRate_{w(1)} + TranspRate_{w(2)} + TranspRate_{w(3)} + TranspRate_{w(4)}$$

donde las tasas de transpiración $TranspRate_{w(i)}$ (mm d⁻¹) se calculan en función de la Tasa de transpiración real están dadas por las ecuaciones:

$$TranspRate_{w(1)} = TasaEvapReal / f(x_{PRaiz}, w(1)) f_{estresW(1)} \cdot Z_{rt1} \cdot f_{AguaContw(1)}$$

$$TranspRate_{w(2)} = TasaEvapReal / f(x_{PRaiz}, w(2)) f_{estresW(2)} \cdot Z_{rt2} \cdot f_{AguaContw(2)}$$

$$TranspRate_{w(1)} = TasaEvapReal / f(x_{PRaiz}, w(3)) f_{estresW(3)} \cdot Z_{rt3} \cdot f_{AguaContw(3)}$$

$$TranspRate_{w(4)} = TasaEvapReal / f(x_{PRaiz}, w(4)) f_{estresW(4)} \cdot Z_{rt4} \cdot f_{AguaContw(4)}$$

La evaporación potencial total acumulada ($x_{tpotevap}$, mm) es calcula mediante la ecuación:

$$\frac{dx_{tpotevap}}{dt} = TasaEvapPot$$

donde:

$$TasaEvapPot = \exp(-0.5x_{IAF})(TasaEvapRad + TasaEvapAire)$$

La evaporación real total acumulada ($x_{trealevap}$, mm) es calculada mediante la ecuación:

$$\frac{dx_{trealevap}}{dt} = insw(LLuvia - 0.5, EvapRateNoLluvia, EvapRateLLuvia)$$

En la Figura 3.5 se muestra un esquema de las variables de estado y los principales componentes del modelo SUCROS972 para crecimiento limitado por agua.

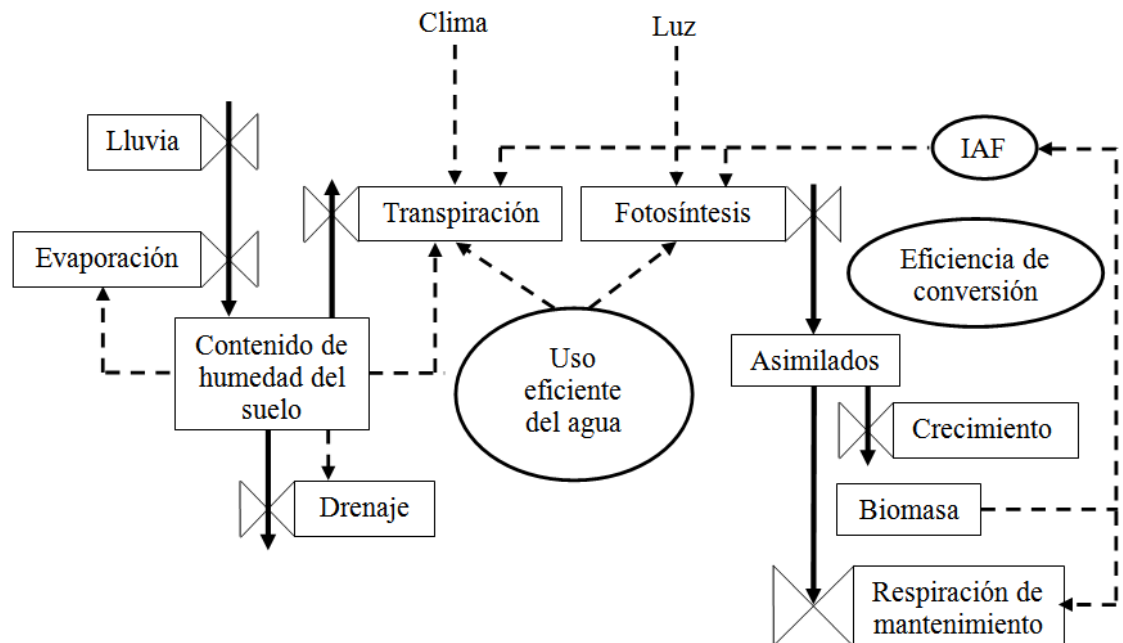


Figura 3.5. Diagrama relacional que muestran los principales componentes del modelo SUCROS972. (De acuerdo con van Laar et al., 1997).

Se usaron las implementaciones computacionales de los tres modelos: SUCROS971, SUCROS972, generados por el grupo de ecología de cultivos y malezas (Van Laar et al, 1997) de la Universidad de Wageningen, Holanda y LINTUL1 generado por la Escuela de Graduados C. T. de Wit de Ecología de la Producción de la Universidad de Wageningen, Holanda, 1997. Estos programas están escritos en el ambiente de simulación Fortran Simulation Translator (FST). Para este trabajo se utilizaron implementaciones de los modelos en el ambiente Matlab.

3.5. Análisis de sensibilidad local

El análisis de sensibilidad se llevó a cabo mediante la implementación de los modelos en el ambiente MATLAB/SIMULINK, se usó el método de integración Euler (ode1 de Matlab) con tamaño de paso de integración fijo: $\Delta t = 1$ día. Se realizaron simulaciones para la variable de salida peso seco total, perturbando los valores de los parámetros de los modelos, incrementándolos en un porcentaje pequeño (incremento del 1%, 4%, 5% y 10% al valor nominal) con la finalidad de obtener y comparar los índices de sensibilidad y la función de sensibilidad de cada parámetro. Los valores nominales de los parámetros fueron tomados de literatura.

Se implementó un programa en Matlab con el fin de obtener la función de sensibilidad para cada uno de los parámetros del modelo, se utilizó la función **trapz.m** para calcular una aproximación de la integración numérica mediante el método trapezoidal y de esta manera obtener el índice de sensibilidad para cada parámetro.

3.6. Calibración de los modelos SUCROS971, LINTUL1 y SUCROS972.

Se llevó a cabo una calibración automática de los modelos utilizando el algoritmo de mínimos cuadrados (`lsqnonlin.m`) disponible en MATLAB para ajustar lo más posible las predicciones de los modelos a las observaciones en campo. Para esto se implementaron los modelos en el ambiente MATLAB/SIMULINK, partiendo de un programa principal donde se definen los parámetros, las variables de entrada y las opciones de simulación, un archivo C-MEX S-function es llamado por el programa principal a través de un archivo mdl de SIMULINK donde se implementan las ecuaciones del modelo. Se establecieron límites superior e inferior para los distintos parámetros a calibrar para cada uno de los modelos de crecimiento del cultivo, algunos fueron tomados de literatura. Se compararon los resultados que generó el modelo con los datos obtenidos en campo.

Capítulo 4. RESULTADOS Y DISCUSIÓN

4.1. Análisis de sensibilidad para los modelos de crecimiento potencial.

Se realizaron simulaciones para la variable de salida peso seco total, perturbando los valores de los parámetros de los modelos, incrementándolos en diferentes porcentajes (1%, 4%, 5%, 10%) para obtener los índices de sensibilidad y la función de sensibilidad de cada parámetro. Los valores nominales de los parámetros fueron tomados de la literatura. En el Cuadro 4.1 se muestran los parámetros del modelo SUCROS971, así como los índices de sensibilidad para las simulaciones de la variable peso seco total con los diferentes porcentajes de perturbación.

Cuadro 4.1. Índices de sensibilidad de los parámetros del modelo de crecimiento potencial SUCROS971 para la variable peso seco total.

Parámetro	Valor nominal del parámetro	Índice de Sensibilidad			
		1%	4%	5%	10%
RGRL	0.04	158.4257	1.3543	32.0438	99.2047
EFF	4.43×10^{-6}	37.0416	36.9277	36.8872	36.6834
KDF	0.8244	30.1351	29.889	29.805	29.4033
AMX	0.0033	4.9105	4.7923	4.7516	4.5563
SCP	0.20	-2.0868	-2.0863	-2.089	-2.1043
SLA	0.001	0.3719	0.3823	0.383	0.0349
LAICR	4.0	-0.0139	-0.0035	-0.0028	-0.0014
FRTRL	0.20	-0.0139	-0.0035	-0.0028	-0.0014
FRDR	1.0	-0.0139	-0.0035	-0.0028	-0.0014
CONVL	0.947	-0.0139	-0.0035	-0.0028	-0.0014

En el Cuadro 4.1 se puede observar que los parámetros más sensibles al modelo son la tasa de crecimiento relativo durante la fase de crecimiento exponencial del área foliar (RGRL), seguido de la eficiencia del uso de luz por las hojas Individuales (EFF), el coeficiente de extinción de la radiación del follaje

(KDF), la tasa potencial de asimilación de CO_2 a saturación de luz para hojas individuales (AMX), el coeficiente de dispersión de las hojas para la radiación fotosintéticamente activa (SCP) y el área foliar específica (SLA). Los demás parámetros con índice de sensibilidad menor a este último, no se consideran sensibles al modelo, y no se toman en cuenta para la calibración. Se puede observar que el valor del índice de sensibilidad del parámetro RGRL varía considerablemente en cada uno de los porcentajes de perturbación en comparación con el resto de los parámetros.

En las Figuras 4.1-4.6 se muestran las simulaciones obtenidas para la variable peso seco total con y sin perturbación, así como la función de sensibilidad para los parámetros más sensibles al modelo SUCROS971, correspondientes a una variación del 1%. Están ordenadas de acuerdo con el valor del índice de sensibilidad para cada parámetro, tomándose RGRL como el parámetro más sensible al modelo y SLA como el último parámetro tomado en cuenta, ya que el resto de los parámetros mostrados en el Cuadro 4.1 se considera que tienen un índice de sensibilidad bajo.

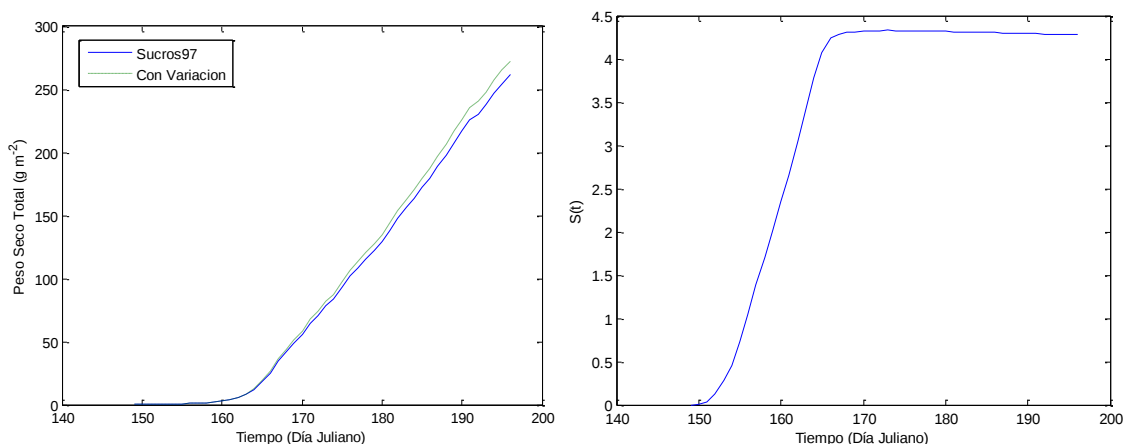


Figura 4.1. Simulaciones y función de sensibilidad para RGRL (SUCROS971).

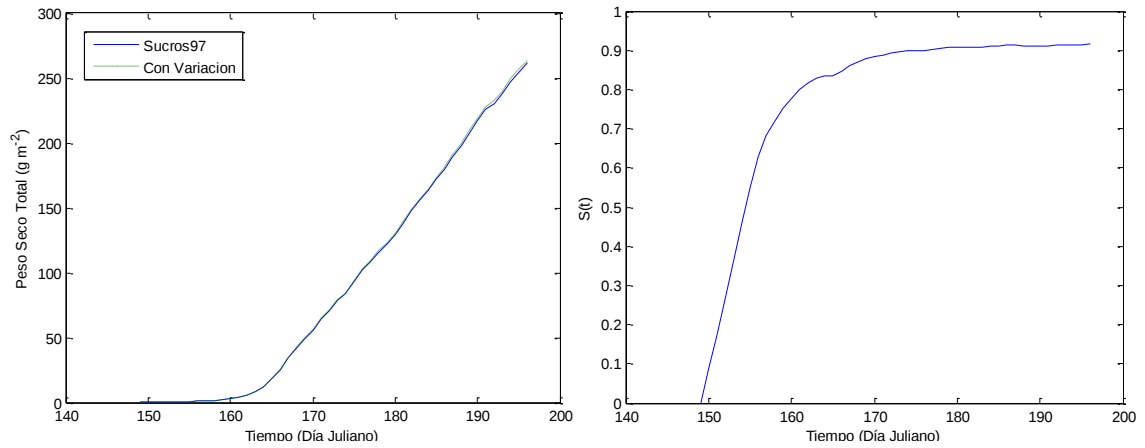


Figura 4.2. Simulaciones y función de sensibilidad para EFF (SUCROS971).

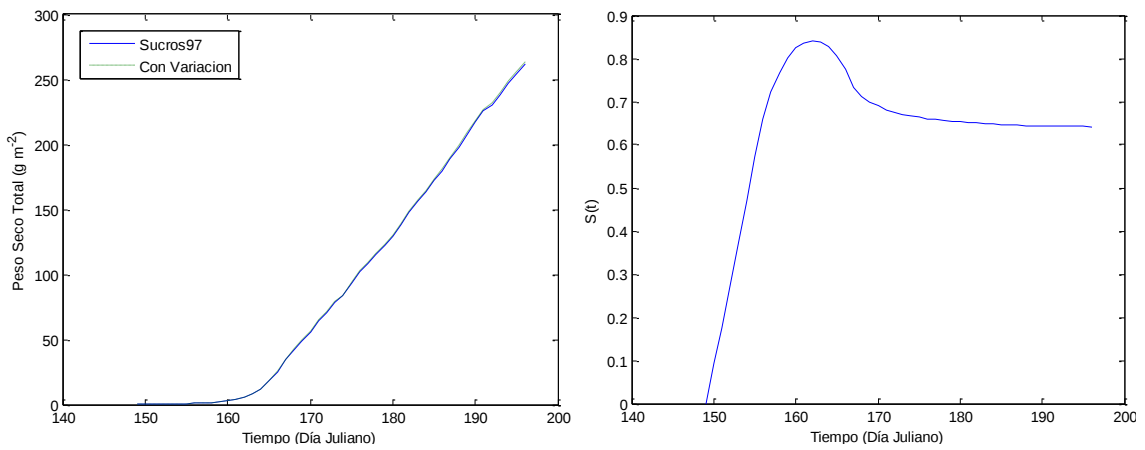


Figura 4.3. Simulaciones y función de sensibilidad para KDF (SUCROS971).

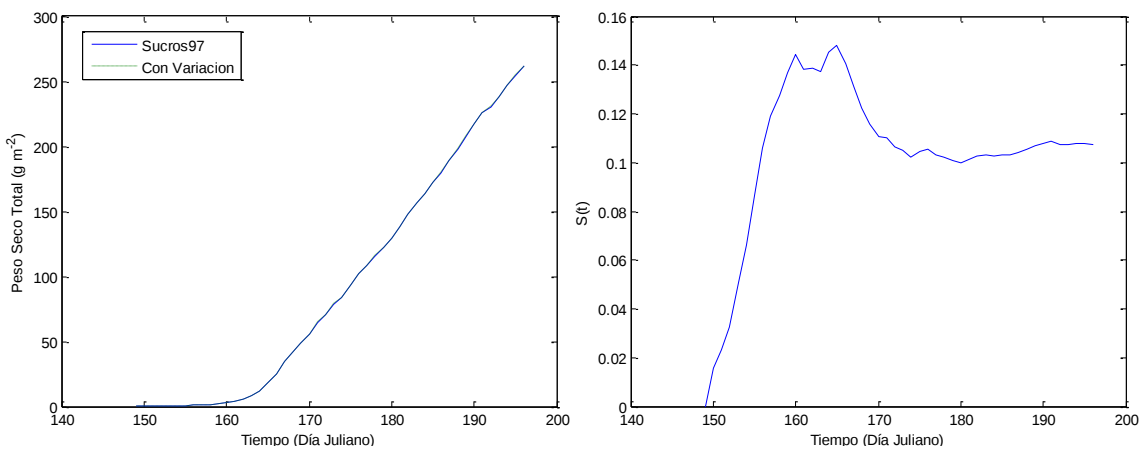


Figura 4.4. Simulaciones y función de sensibilidad para AMX (SUCROS971).

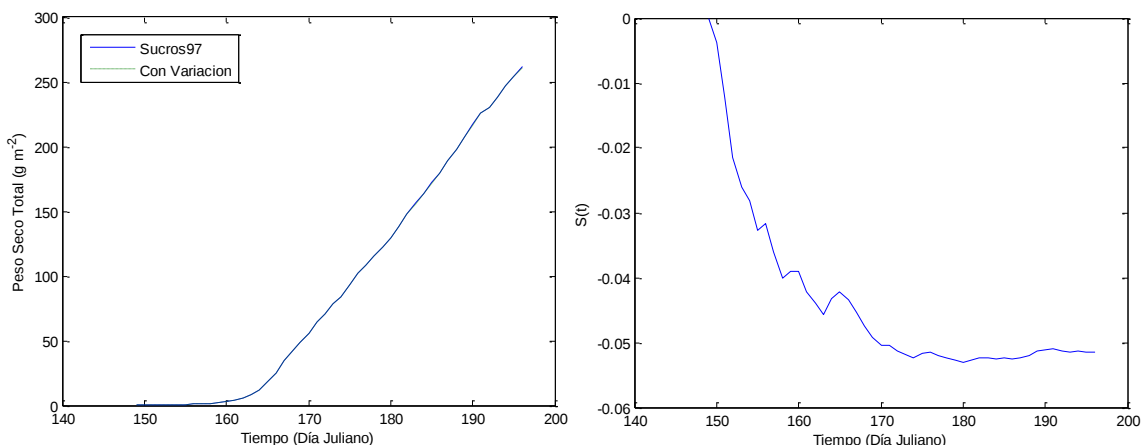


Figura 4.5. Simulaciones y función de sensibilidad para SCP (SUCROS971).

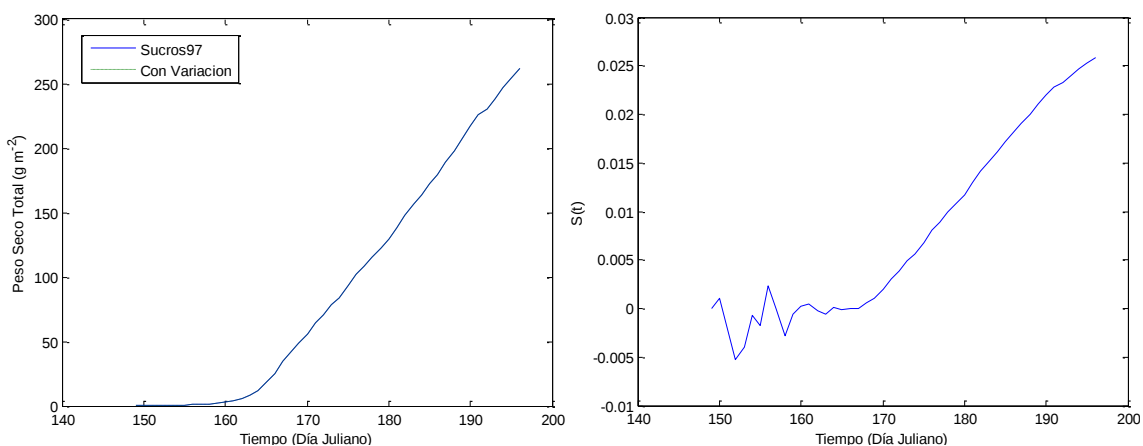


Figura 4.6. Simulaciones y función de sensibilidad para SLA (SUCROS971).

En el Cuadro 4.2 se muestran los parámetros del modelo LINTUL1, así como los índices de sensibilidad para las simulaciones de la variable peso seco total con los diferentes porcentajes de perturbación.

Cuadro 4.2. Índices de sensibilidad de los parámetros del modelo simple de crecimiento potencial LINTUL1 para la variable peso seco total.

Parámetro	Valor nominal del parámetro	Índice de Sensibilidad			
		1%	4%	5%	10%
RGR1	0.009	52.7049	54.3376	54.8928	57.7922
LUE	3.0	39.4185	39.4995	39.5216	39.6294
K	0.6	39.2266	39.2975	39.3161	39.4058
SLA	0.022	1.5835	1.6158	1.6214	1.6457
LAICR	4.0	-0.0258	-0.0064	-0.0052	-0.0026
RDRSHM	0.03	-0.0258	-0.0064	-0.0052	-0.0026
TSUMAN	1110	-0.0258	-0.0064	-0.0052	-0.0026

Al igual que en el modelo SUCROS971 los parámetros más sensibles al modelo LINTUL1 son la tasa de crecimiento relativo durante la fase de crecimiento exponencial del área foliar (RGRL), seguido de la eficiencia del uso de la luz (materia seca producida por unidad de radiación fotosintéticamente activa interceptada) (LUE), el coeficiente de extinción para la radiación fotosintéticamente activa (K) y el área foliar específica (SLA). El resto de los parámetros mostrados en el Cuadro 4.2 no son tomados en cuenta para ser calibrados, ya que su índice de sensibilidad es pequeño.

En las Figuras 4.7-4.10 se muestran las simulaciones para la variable peso seco total con y sin perturbación, así como la función de sensibilidad para los parámetros más sensibles al modelo LINTUL1, correspondientes a una variación del 1%. Están ordenadas de acuerdo con el valor del índice de sensibilidad para cada parámetro, tomándose RGRL como el parámetro más sensible al modelo y SLA como el último parámetro tomado en cuenta para ser calibrados.

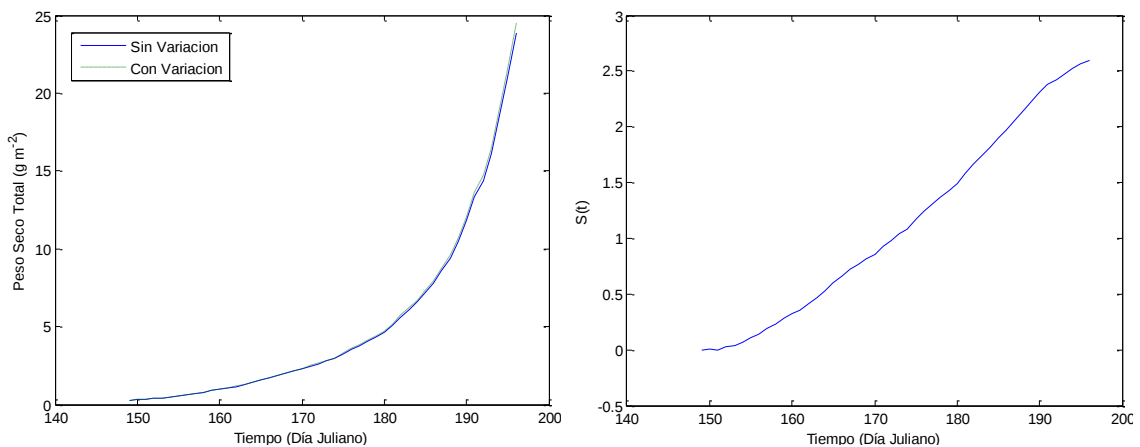


Figura 4.7. Simulaciones y función de sensibilidad para RGRL (LINTUL1).

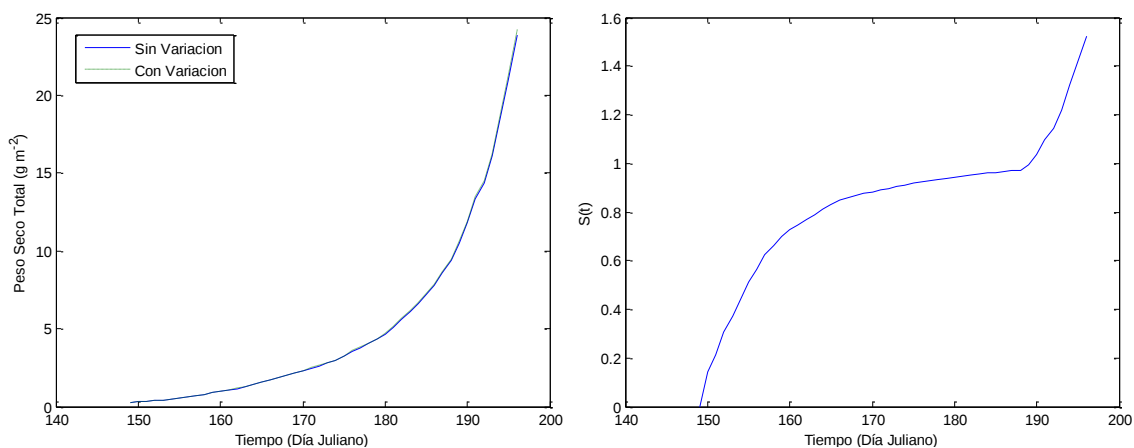


Figura 4.8. Simulaciones y función de sensibilidad para LUE (LINTUL1).

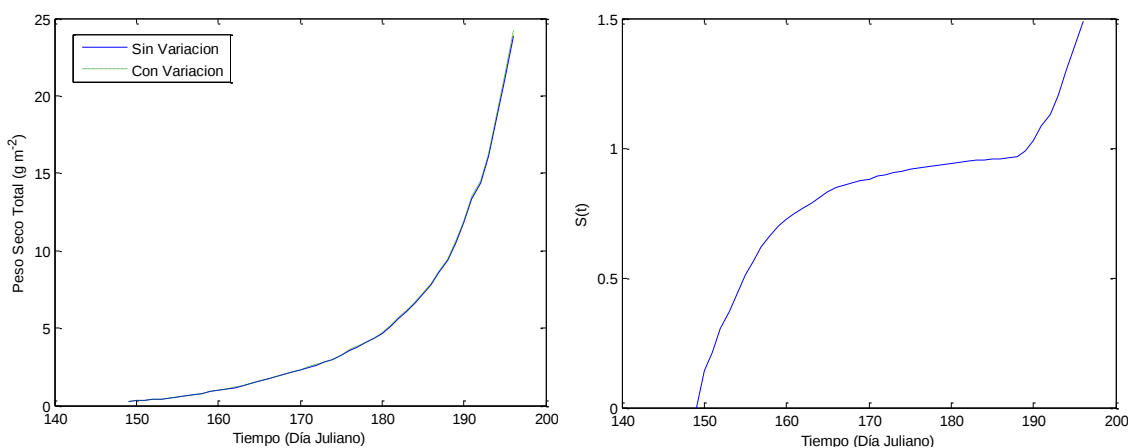


Figura 4.9. Simulaciones y función de sensibilidad para K (LINTUL1).

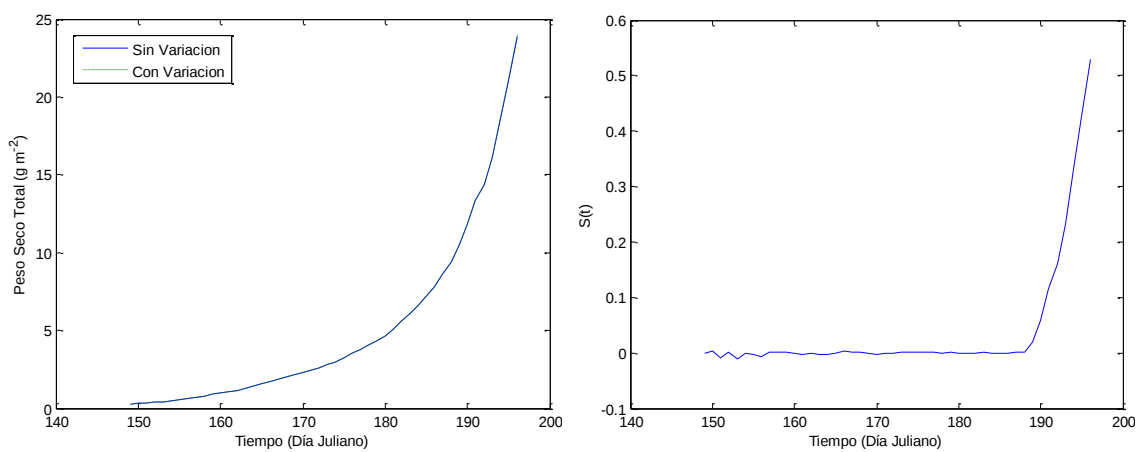


Figura 4.10. Simulaciones y función de sensibilidad para SLA (LINTUL1).

4.2. Calibración automática de los modelos de crecimiento potencial.

4.2.1. Calibración del modelo SUCROS971.

Se hizo una calibración automática variando dentro de un rango pequeño los valores de los parámetros que hacen más sensible al modelo, con la finalidad de ajustar lo más posible las predicciones del modelo a las observaciones en campo, se llevaron a cabo varias ejecuciones del programa de calibración, asignando distintos valores semilla a los parámetros, ya que la función de minimización de los cuadrados medios del error en ocasiones converge a un mínimo local y no al mínimo global deseado.

En los Cuadros 4.3-4.8 se muestran varias ejecuciones del modelo para la variable de salida peso seco total, así como los límites inferior y superior para cada parámetro y sus valores correspondientes a cada ejecución. Los valores calibrados para los parámetros son los correspondientes al valor más bajo de la función a minimizar, la cual fue la suma de los cuadrados de los errores. En la Figuras 4.11-4.16 se muestran las simulaciones con y sin calibración del crecimiento potencial del cultivo de tomate de cáscara para la variable de salida peso seco total y las diferentes variables de estado, usando el método de mínimos cuadrados no lineales (función *lsqnonlin.m*), con intervalos de confianza con un nivel de significancia $\alpha=0.05$ para los valores medidos en campo. Las simulaciones para las variables no calibradas se presentan en el Apéndice A.

Cuadro 4.3. Resultados de la calibración automática del modelo SUCROS971 para la variable peso seco total.

Parámetro	Rango del parámetro	Calibración								
		1	2	3	4	5	6	7	8	9
RGR	0.001-0.04	0.0359	0.0353	0.0357	0.0366	0.0354	0.0363	0.0351	0.0358	0.0367
EFF	12.5×10^{-7} - 2.375×10^{-5}	3.4539×10^{-6}	5.6705×10^{-6}	7.5063×10^{-6}	4.8102×10^{-6}	4.88×10^{-6}	1.2892×10^{-5}	4.7949×10^{-6}	6.3929×10^{-6}	7.6258×10^{-6}
KDF	0.25-0.95	0.9332	0.6953	0.4813	0.6195	0.7769	0.25	0.85	0.5496	0.3845
AMX	1×10^{-3} - 1.22×10^{-3}	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
SCP	0.1-0.3	0.1159	0.1018	0.1	0.1016	0.1164	0.1	0.1	0.1001	0.1001
SLA	0.002-0.042	0.0417	0.0202	0.0184	0.0269	0.0235	0.0178	0.0228	0.0203	0.0211
f(x)		1094.04 *	1199.58	1211.59	1183.16	1183.56	1210.41	1179.92	1205.59	1208.19

* Valor mínimo de la función de calibración.

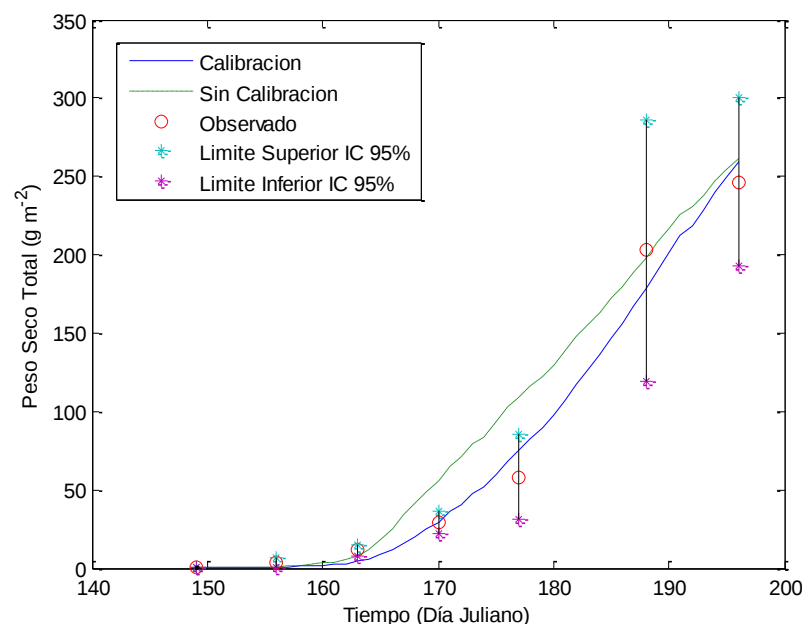


Figura 4.11. Simulación del crecimiento, sin calibración, y con calibración para la variable peso seco total (SUCROS971).

Cuadro 4.4. Resultados de la calibración automática del modelo SUCROS971 para la variable peso seco de hojas.

Parámetro	Rango del parámetro	Calibración								
		1	2	3	4	5	6	7	8	9
RGR	0.001-0.04	0.0277	0.0277	0.0277	0.0277	0.0277	0.0277	0.0277	0.0277	0.0277
EFF	12.5×10^{-7} - 2.375×10^{-5}	1.8867×10^{-5}	1.8867×10^{-5}	1.8919×10^{-5}	1.8867×10^{-5}	1.8867×10^{-5}	1.8867×10^{-5}	1.8867×10^{-5}	1.8867×10^{-5}	1.8867×10^{-5}
KDF	0.25-0.95	0.95	0.95	0.9499	0.95	0.95	0.95	0.95	0.95	0.95
AMX	1×10^{-3} - 1.22×10^{-3}	0.0012	0.0012	0.0012	0.0012	0.0012	0.0012	0.0012	0.0012	0.0012
SCP	0.1-0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3
SLA	0.002-0.042	0.0123	0.0123	0.0123	0.0123	0.0123	0.0123	0.0123	0.0123	0.0123
f(x)		26.7741	26.7741	26.7742	26.7741	26.7741	26.7741	26.7741	26.7741	26.7741 *

* Valor mínimo de la función de calibración.

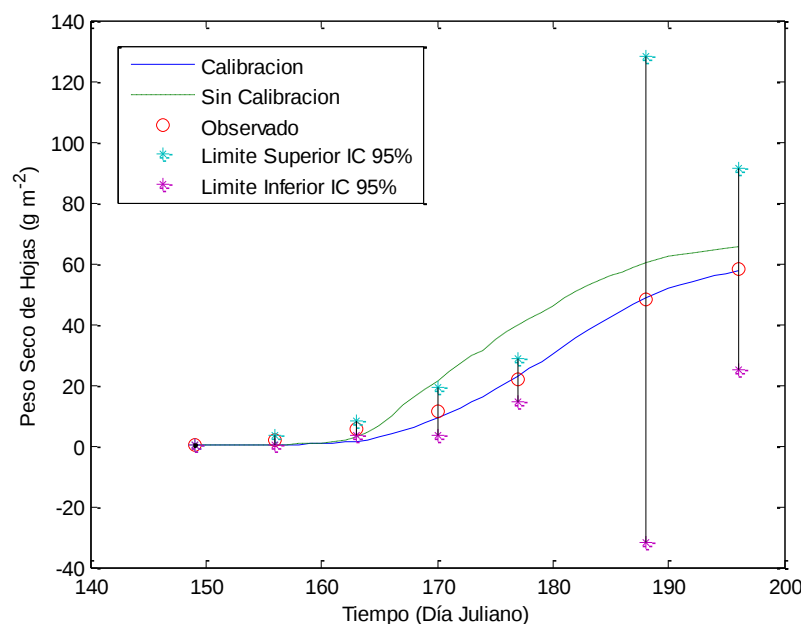


Figura 4.12. Simulación del crecimiento, sin calibración, y con calibración para la variable peso seco de hojas (SUCROS971).

Cuadro 4.5. Resultados de la calibración automática del modelo SUCROS971 para la variable peso seco de tallos.

Parámetro	Rango del parámetro	Calibración								
		1	2	3	4	5	6	7	8	9
RGR	0.001-0.04	0.0364	0.0376	0.0389	0.0374	0.0379	0.0389	0.04	0.0388	0.04
EFF	12.5×10^{-7} - 2.375×10^{-5}	9.3223×10^{-6}	7.9245×10^{-6}	5.7965×10^{-6}	7.6952×10^{-6}	7.3019×10^{-6}	6.2906×10^{-6}	5.6225×10^{-6}	5.5858×10^{-6}	5.8320×10^{-6}
KDF	0.25-0.95	0.8875	0.7981	0.8734	0.8502	0.7986	0.7966	0.95	0.9289	0.9076
AMX	1×10^{-3} - 1.22×10^{-3}	0.0012	0.0012	0.0012	0.0012	0.0012	0.0011	0.0012	0.0012	0.0012
SCP	0.1-0.3	0.2996	0.2999	0.2995	0.3	0.3	0.2996	0.3	0.3	0.3
SLA	0.002-0.042	0.003	0.0034	0.0046	0.0034	0.0035	0.0043	0.0065	0.0046	0.0063
f(x)		10.8238	10.9372	10.476	10.7474	10.7906	10.8671	7.66031 *	10.2222	7.78332

* Valor mínimo de la función de calibración.

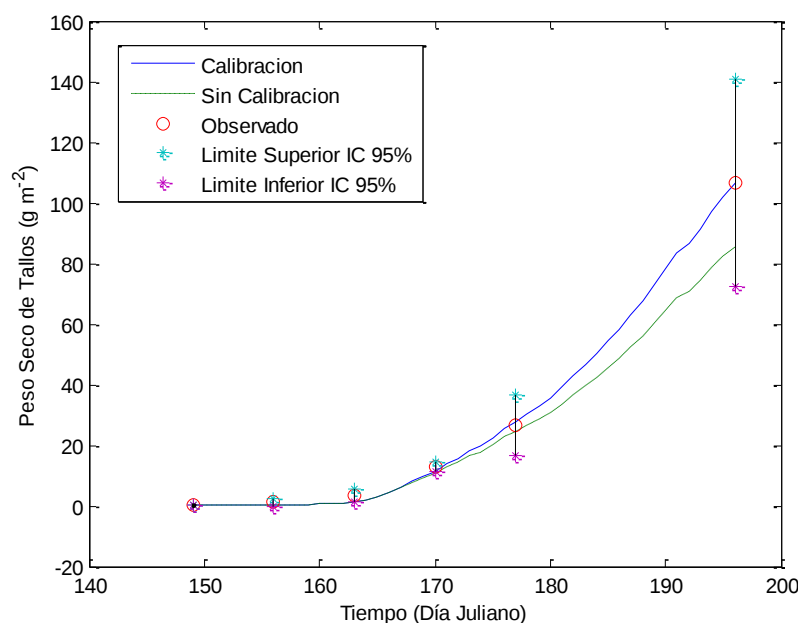


Figura 4.13. Simulación del crecimiento, sin calibración, y con calibración para la variable peso seco de tallos (SUCROS971).

Cuadro 4.6. Resultados de la calibración automática del modelo SUCROS971 para la variable peso seco de raíces.

Parámetro	Rango del parámetro	Calibración								
		1	2	3	4	5	6	7	8	9
RGR	0.001-0.04	0.0389	0.0231	0.0231	0.0231	0.0231	0.0231	0.0231	0.0389	0.0231
EFF	12.5×10^{-7} - 2.375×10^{-5}	1.5902×10^{-6}	2.3750×10^{-5}	2.3750×10^{-5}	2.3750×10^{-5}	2.3750×10^{-5}	2.3750×10^{-5}	2.3750×10^{-5}	2.3284×10^{-6}	2.3750×10^{-5}
KDF	0.25-0.95	0.5872	0.9500	0.9500	0.9500	0.9500	0.9500	0.9500	0.3573	0.9500
AMX	1×10^{-3} - 1.22×10^{-3}	0.0011	0.0012	0.0012	0.0012	0.0012	0.0012	0.0012	0.0011	0.0012
SCP	0.1-0.3	0.2785	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.2797	0.3000
SLA	0.002-0.042	0.027	0.0042	0.0042	0.0042	0.0042	0.0042	0.0042	0.0377	0.0042
f(x)		1.88407	1.24787	1.24787	1.24787	1.24787	1.24787	1.24787	2.05022	1.24787 *

* Valor mínimo de la función de calibración.

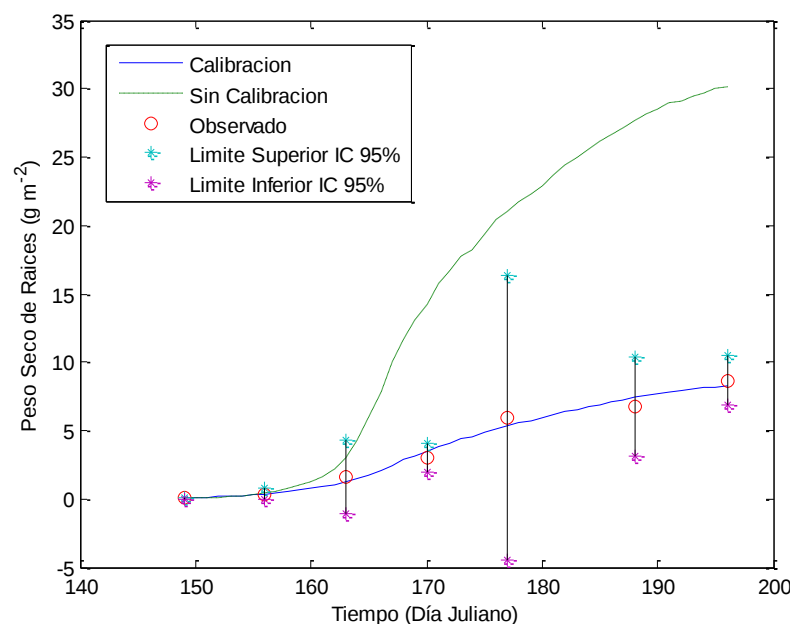


Figura 4.14. Simulación del crecimiento, sin calibración, y con calibración para la variable peso seco de raíces (SUCROS971).

Cuadro 4.7. Resultados de la calibración automática del modelo SUCROS971 para la variable peso seco de frutos.

Parámetro	Rango del parámetro	Calibración								
		1	2	3	4	5	6	7	8	9
RGR	0.001-0.04	0.0264	0.0264	0.0264	0.0264	0.0389	0.0264	0.0264	0.0264	0.0264
EFF	12.5×10^{-7} - 2.375×10^{-5}	4.5122×10^{-6}	4.5122×10^{-6}	4.5122×10^{-6}	4.5122×10^{-6}	6.2740×10^{-6}	4.5122×10^{-6}	4.5122×10^{-6}	4.5122×10^{-6}	4.5122×10^{-6}
KDF	0.25-0.95	0.95	0.95	0.95	0.95	0.4747	0.95	0.95	0.95	0.95
AMX	1×10^{-3} - 1.22×10^{-3}	0.0012	0.0012	0.0012	0.0012	0.0012	0.0012	0.0012	0.0012	0.0012
SCP	0.1-0.3	0.1	0.1	0.1	0.1	0.1977	0.1	0.1	0.1	0.1
SLA	0.002-0.042	0.042	0.042	0.042	0.042	0.0086	0.042	0.042	0.042	0.042
f(x)		29.8626	29.8626	29.8626	29.8626	228.066	29.8626	29.8626	29.8626	29.8626 *

* Valor mínimo de la función de calibración.

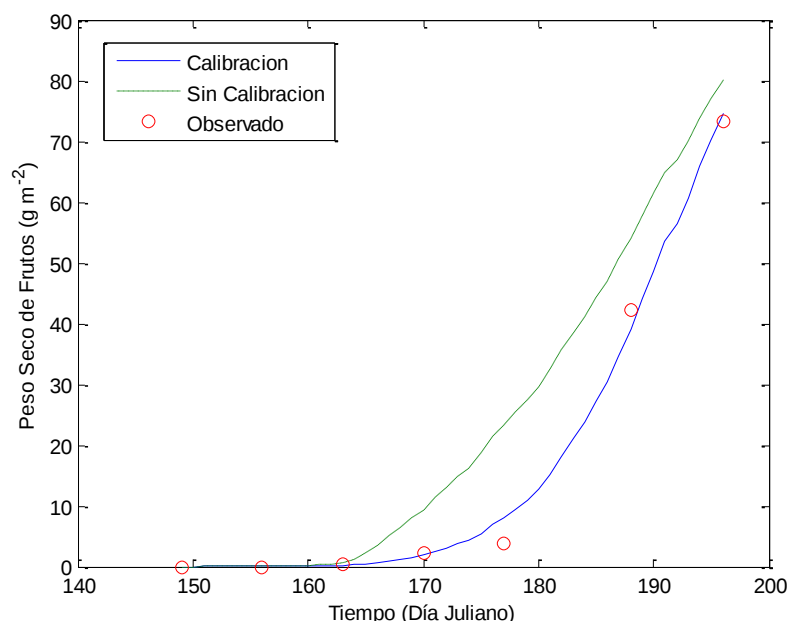


Figura 4.15. Simulación del crecimiento, sin calibración, y con calibración para la variable peso seco de frutos (SUCROS971).

Cuadro 4.8. Resultados de la calibración automática del modelo SUCROS971 para la variable índice de área foliar.

Parámetro	Rango del parámetro	Calibración								
		1	2	3	4	5	6	7	8	9
RGR	0.001-0.04	0.0298	0.0298	0.0298	0.0298	0.0298	0.0298	0.0298	0.0389	0.0298
EFF	12.5×10^{-7} - 2.375×10^{-5}	6.2225×10^{-6}	6.2225×10^{-6}	6.2225×10^{-6}	6.2225×10^{-6}	6.2225×10^{-6}	6.3429×10^{-6}	6.2225×10^{-6}	1.2628×10^{-5}	3.8841×10^{-6}
KDF	0.25-0.95	0.9499	0.9499	0.9499	0.9499	0.9499	0.95	0.9499	0.6316	0.9438
AMX	1×10^{-3} - 1.22×10^{-3}	0.0012	0.0012	0.0012	0.0012	0.0012	0.0012	0.0012	0.0012	0.001
SCP	0.1-0.3	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1912	0.1136
SLA	0.002-0.042	0.0285	0.0285	0.0285	0.0285	0.0285	0.0282	0.0285	0.0082	0.0420
f(x)		0.117776	0.117776	0.117776	0.117776	0.117776	0.117776	0.117776 *	0.664477	0.117951

* Valor mínimo de la función de calibración.

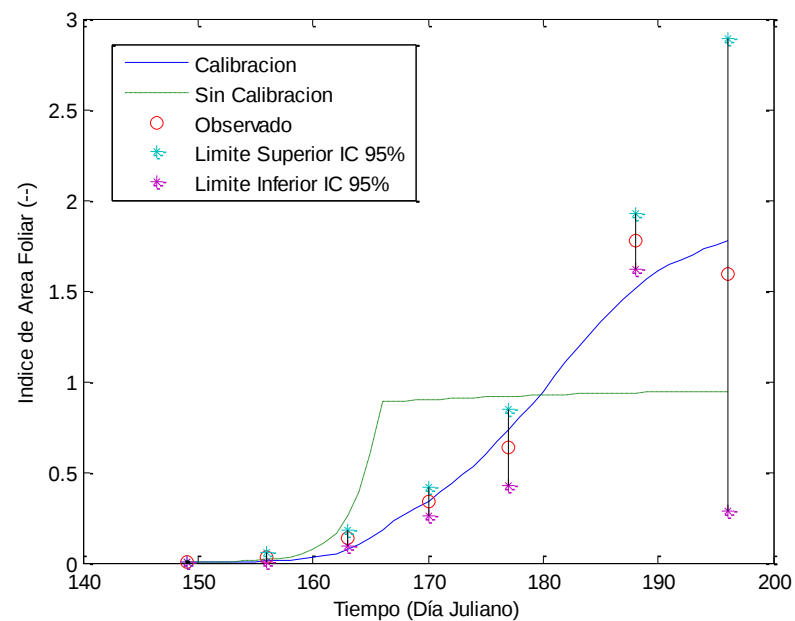


Figura 4.16. Simulación del crecimiento, sin calibración, y con calibración para la variable índice de área foliar (SUCROS971).

En las figuras anteriores se puede observar que después de la calibración, utilizando el método de mínimos cuadrados, el modelo SUCROS971 es capaz de predecir en forma apropiada el comportamiento de la biomasa total, biomasa de hojas, biomasa de tallos, biomasa de raíces, biomasa de frutos y el índice de área foliar del cultivo. En el Cuadro 4.9 se presentan la raíz cuadrada del cuadrado medio del error (RMSE) y el coeficiente de determinación (R^2) de la ecuación de regresión resultante entre los datos medidos y las predicciones obtenidas mediante el modelo.

Cuadro 4.9. Raíz cuadrada del cuadrado medio del error (RMSE) y coeficiente de determinación (R^2) de la ecuación de regresión entre variables medidas y estimadas obtenidos de la calibración del modelo SUCROS971.

Variable Calibrada	Biomasa total		Hojas		Tallos		Raíces		Frutos		IAF	
	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE
Biomasa total	0.9828	12.48	0.9861	2.775	0.9953	7.606	0.9678	8.349	0.982	8.053	0.944	0.6525
Hojas	0.9729	18.5095	0.9951	1.9557	0.9871	5.8676	0.9526	7.8448	0.9909	11.6183	0.9551	0.5101
Tallos	0.9697	39.3979	0.9579	14.5505	0.9994	1.1390	0.9748	15.5295	0.9603	15.9735	0.7182	0.3719
Raíces	0.9760	85.6841	0.9720	20.0241	0.9993	33.8982	0.9818	0.4222	0.9719	20.9628	0.8058	0.8564
Frutos	0.9627	34.9335	0.9939	10.6060	0.9772	13.6755	0.9127	3.3889	0.9942	2.0655	0.9589	0.1697
IAF	0.9741	20.1184	0.9955	2.1926	0.9847	6.3895	0.9396	7.6464	0.9927	12.9191	0.9647	0.1297

El Cuadro 4.9 indica que solo en el caso de las calibraciones para tallos y raíces, donde el valor del coeficiente de determinación para el índice de área foliar (IAF) es de 0.7182 y 0.8058 respectivamente, la predicción de la variable no presenta una tendencia similar a la observada, lo cual se puede apreciar también en las figuras de las simulaciones para las variables de estado no calibradas presentadas en el Apéndice A.

Se puede apreciar que el valor de R^2 es más elevado en las variables para las cuales se realizó la calibración, presentando valores entre 0.98 y 0.99, lo que indica que hay una tendencia similar entre predicciones y observaciones. De la misma manera se puede observar que los valores más bajos de RMSE se presentan en las variables para las cuales se realizó la calibración, esto indica un buen ajuste entre las curvas estimadas y las observadas, lo cual se puede ver en las figuras presentadas en Apéndice A.

En el Cuadro 4.10 se muestran los valores de los parámetros para cada una de las variables calibradas. Los valores cambiaron de acuerdo a cada variable calibrada, ya que es posible que existan varias soluciones al problema de optimización en cada calibración, es decir, que el problema de optimización sea multimodal, lo cual puede ser investigado utilizando algoritmos de optimización más sofisticados, como lo son los métodos globales de optimización.

Cuadro 4.10. Valores de los parámetros del modelo SUCROS971 después de la calibración usando el método de mínimos cuadrados.

Variable calibrada	Parámetro					
	RGRL	EFF	KDF	AMX	SCP	SLA
Biomasa Total	0.0359	3.4539×10^{-6}	0.9332	0.001	0.1159	0.0417
Hojas	0.0277	1.8867×10^{-5}	0.95	0.0012	0.3	0.0123
Tallos	0.04	5.6225×10^{-6}	0.95	0.0012	0.3	0.0065
Raíces	0.0231	2.3750×10^{-5}	0.95	0.0012	0.3	0.0042
Frutos	0.0264	4.5122×10^{-6}	0.95	0.0012	0.1	0.042
IAF	0.0298	6.2225×10^{-6}	0.9499	0.0012	0.1	0.0285
Desviación estándar	0.0064	8.6383×10^{-6}	0.00685	8.165×10^{-5}	0.1068	0.0172

4.2.2. Calibración del modelo simple de crecimiento potencial LINTUL1.

Se llevó a cabo una calibración automática variando dentro de un rango pequeño los valores de los parámetros que hacen más sensible al modelo, con la finalidad de ajustar los más posible las predicciones del modelo a las observaciones en campo, de la misma manera que en la calibración para SUCROS971 se llevaron a cabo varias ejecuciones del programa de calibración asignando distintos valores semilla a los parámetros.

En los Cuadros 4.11-4.15 se muestran varias ejecuciones del modelo para la variable de salida peso seco total, así como los límites inferior y superior para cada parámetro y sus valores correspondientes a cada ejecución. Los valores calibrados para los parámetros son los correspondientes al valor más bajo de la función de calibración $f(x)$. En la figuras 4.17-4.21 se muestran las simulaciones con y sin calibración del crecimiento potencial del cultivo de tomate de cáscara para la variable de salida peso seco total y las diferentes variables de estado, usando el método de mínimos cuadrados no lineales (función *lsqnonlin.m*), con intervalos de confianza con un nivel de significancia $\alpha=0.05$ para los valores medidos en campo. Las simulaciones para las variables no calibradas se presentan en el Apéndice B.

Cuadro 4.11. Resultados de la calibración automática del modelo LINTUL1 para la variable peso seco total.

Parámetro	Rango del parámetro	Calibración								
		1	2	3	4	5	6	7	8	9
RGR	0.001-0.04	0.0349	0.0221	0.0389	0.0271	0.0349	0.0321	0.0349	0.0251	0.0261
LUE	0.5-5.5	1.1782	2.5084	1.3955	2.0808	1.0254	4.625	1.0254	2.525	2.4168
K	0.25-0.95	0.7535	0.6774	0.5477	0.5834	0.84	0.25	0.84	0.5347	0.5225
SLA	0.002-0.042	0.0181	0.002	0.0118	0.0035	0.0334	0.002	0.0334	0.002	0.0029
f(x)		1422.06	3726.04	1504.42	1822.91	1316.91	1441.5	1316.91 *	2255.59	2053.1

* Valor mínimo de la función de calibración.

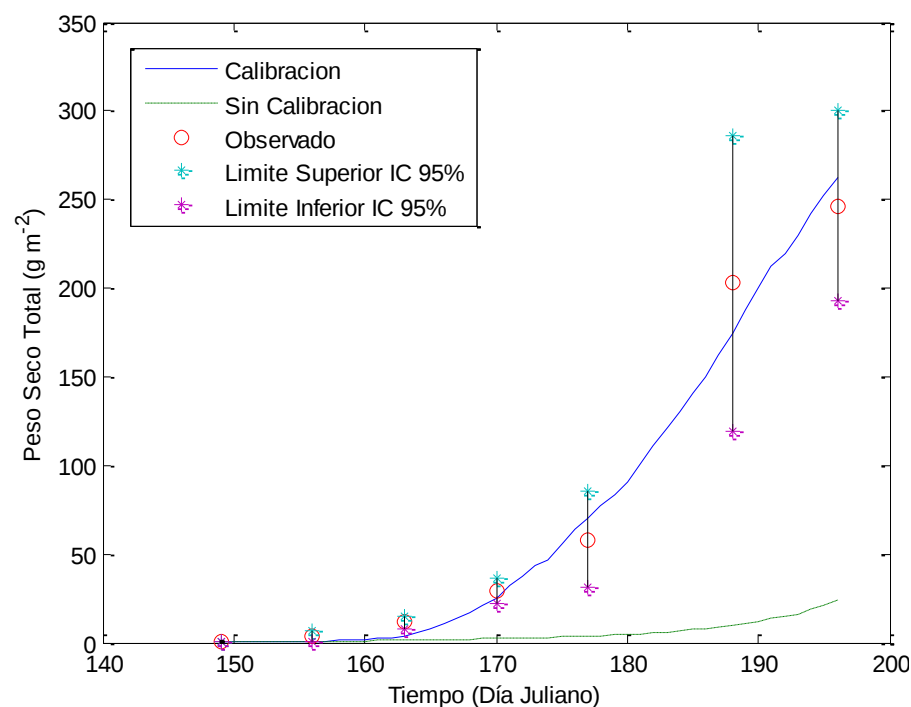


Figura 4.17. Simulación del crecimiento, sin calibración, y con calibración para la variable peso seco total (LINTUL1).

Cuadro 4.12. Resultados de la calibración automática del modelo LINTUL1 para peso seco de hojas.

Parámetro	Rango del parámetro	Calibración								
		1	2	3	4	5	6	7	8	9
RGRL	0.001-0.04	0.0321	0.0349	0.0234	0.0201	0.013	0.0291	0.0251	0.0201	0.0349
LUE	0.5-5.5	0.8007	1.3065	1.3038	1.6321	5.5	1.2053	1.0929	1.5996	1.3066
K	0.25-0.95	0.6641	0.4139	0.6469	0.7155	0.95	0.4932	0.7326	0.7368	0.4140
SLA	0.002-0.042	0.0181	0.002	0.0027	0.0022	0.0020	0.0036	0.002	0.002	0.002
f(x)		212.24	104.045	419.291	560.579	620.311	228.209	325.516	555.591	103.992 *

* Valor mínimo de la función de calibración.

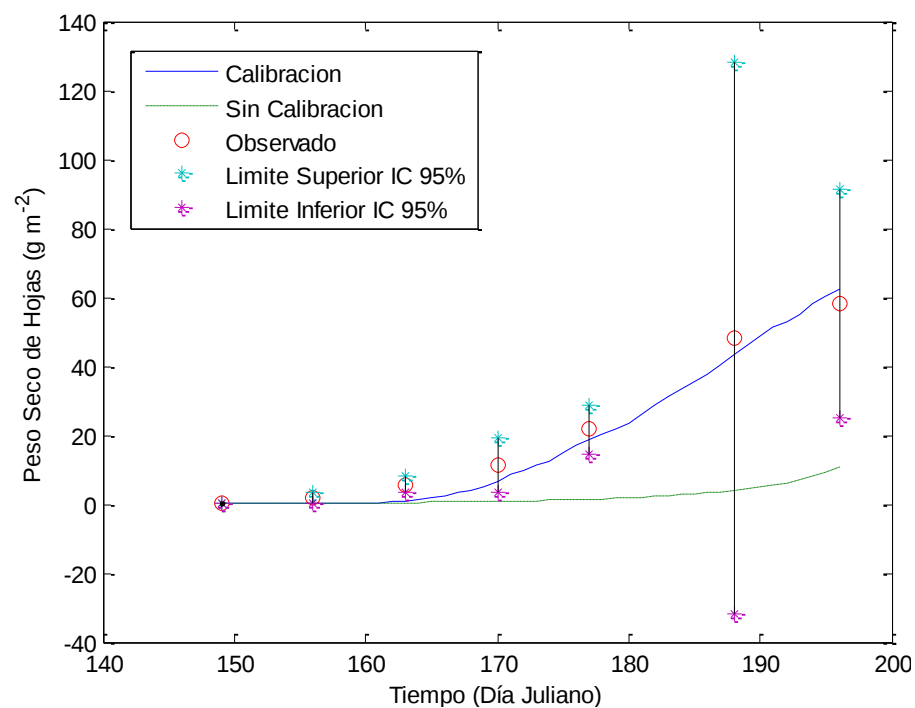


Figura 4.18. Simulación del crecimiento, sin calibración, y con calibración para peso seco de hojas (LINTUL1).

Cuadro 4.13. Resultados de la calibración automática del modelo LINTUL1 para peso seco de tallos.

Parámetro	Rango del parámetro	Calibración								
		1	2	3	4	5	6	7	8	9
RGRL	0.001-0.04	0.0369	0.0389	0.0305	0.0305	0.0349	0.0305	0.0271	0.0251	0.0305
LUE	0.5-5.5	2.315	2.2588	2.513	2.403	1.9247	2.8268	3.5906	2.8586	3.1117
K	0.25-0.95	0.5497	0.5206	0.6077	0.651	0.6969	0.4309	0.3648	0.8959	0.5014
SLA	0.002-0.042	0.0112	0.0131	0.0104	0.0109	0.0205	0.0151	0.0134	0.0023	0.0063
f(x)		35.4876	24.7116 *	149.675	145.524	67.3466	214.4	337.531	230.8	130.392

* Valor mínimo de la función de calibración.

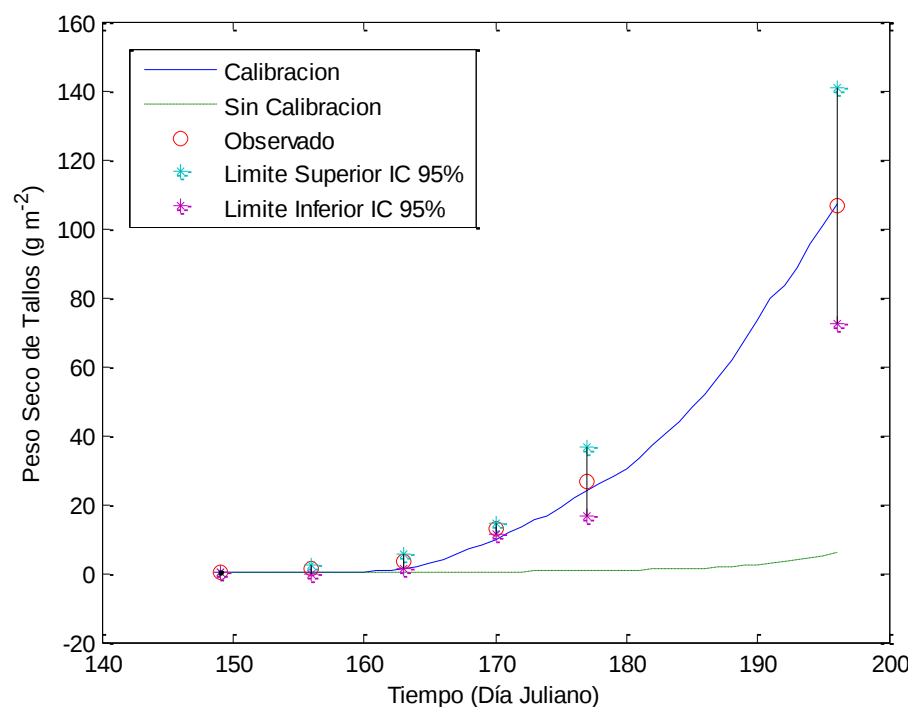


Figura 4.19. Simulación del crecimiento, sin calibración, y con calibración para peso seco de tallos (LINTUL1).

Cuadro 4.14. Resultados de la calibración automática del modelo LINTUL1 para peso seco de raíces.

Parámetro	Rango del parámetro	Calibración								
		1	2	3	4	5	6	7	8	9
RGRl	0.001-0.04	0.007	0.0242	0.0242	0.007	0.007	0.007	0.007	0.007	0.007
LUE	0.5-5.5	5.5	0.6242	0.6218	5.5	5.5	5.5	5.5	5.5	5.5
K	0.25-0.95	0.95	0.2725	0.2732	0.95	0.95	0.95	0.95	0.95	0.95
SLA	0.002-0.042	0.002	0.0023	0.0026	0.002	0.002	0.002	0.002	0.002	0.002
f(x)		7.27119	22.7865	22.8116	7.27119	7.27119	7.27119	7.27119	7.27119	7.27119 *

* Valor mínimo de la función de calibración.

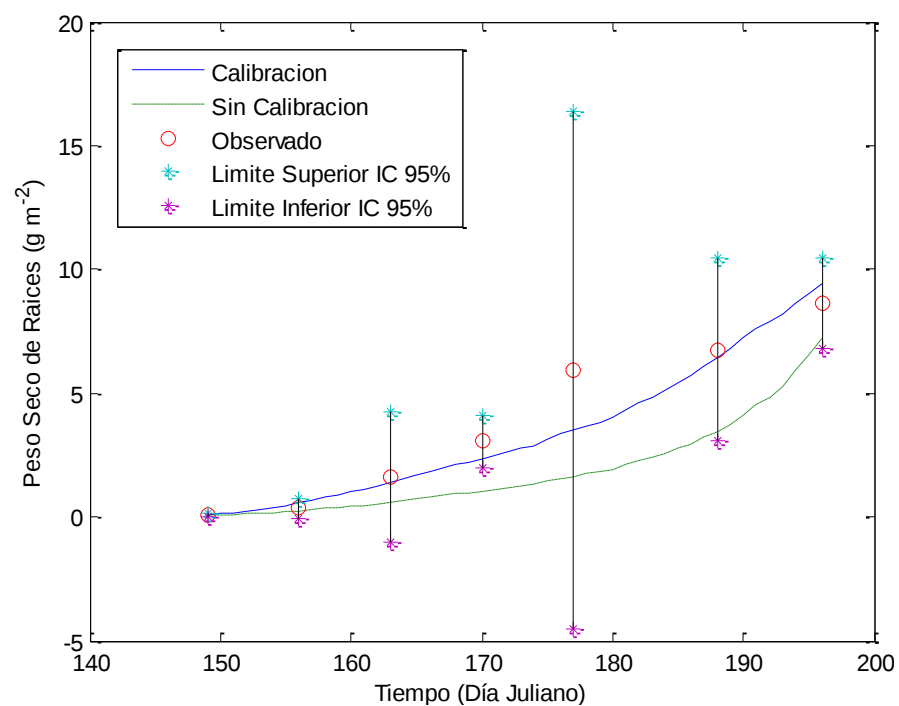


Figura 4.20. Simulación del crecimiento, sin calibración, y con calibración para peso seco de raíces (LINTUL1).

Cuadro 4.15. Resultados de la calibración automática del modelo LINTUL1 para índice de área foliar.

Parámetro	Rango del parámetro	Calibración								
		1	2	3	4	5	6	7	8	9
RGRL	0.001-0.04	0.0242	0.0369	0.0242	0.0349	0.0258	0.0305	0.0227	0.0234	0.0227
LUE	0.5-5.5	2.1172	1.6551	1.7735	0.5	0.5	1.3505	2.4744	1.5516	0.9238
K	0.25-0.95	0.5495	0.4117	0.519	0.5509	0.8716	0.454	0.3806	0.3206	0.4365
SLA	0.002-0.042	0.0086	0.0114	0.0115	0.0414	0.0283	0.0198	0.0112	0.0142	0.0112
f(x)		0.264891 *	0.521453	0.270525	0.654931	0.295801	0.51728	0.359745	0.38452	0.800957

* Valor mínimo de la función de calibración.

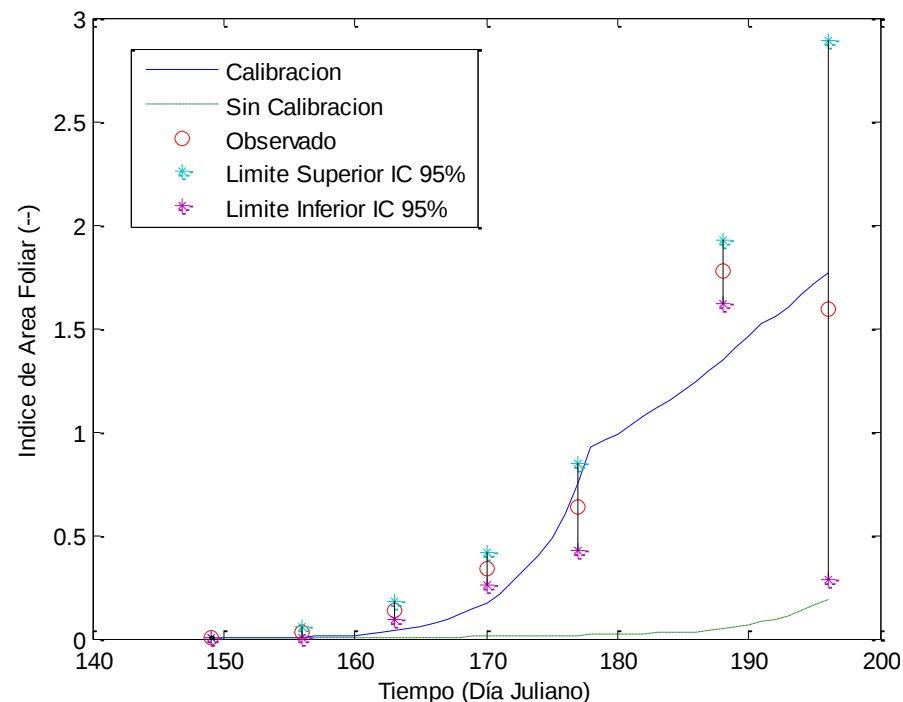


Figura 4.21. Simulación del crecimiento, sin calibración, y con calibración para índice de área foliar (LINTUL1).

En las figuras anteriores se puede observar que después de la calibración, utilizando el método de mínimos cuadrados, el modelo LINTUL1 es capaz de predecir en forma apropiada el comportamiento de la biomasa total, biomasa de hojas, biomasa de tallos, para el caso de la biomasa de raíces y el índice de área foliar presenta dificultades, sin embargo, se presenta un buen ajuste. En el Cuadro 4.16 se presentan la raíz cuadrada del cuadrado medio del error (RMSE) y el coeficiente de determinación (R^2) de la ecuación de regresión resultante entre los datos medidos y las predicciones obtenidas mediante el modelo.

Cuadro 4.16. Raíz cuadrada del cuadrado medio del error (RMSE) y coeficiente de determinación (R^2) de la ecuación de regresión entre variables medidas y estimadas obtenidos de la calibración del modelo LINTUL1.

Variable Calibrada	Biomasa total		Hojas		Tallos		Raíces		IAF	
	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE
Biomasa total	0.9798	13.7158	0.9709	23.866	0.9977	19.8107	0.8942	37.6663	0.9127	1.3361
Hojas	0.9793	55.1414	0.9833	3.8544	0.9988	31.3612	0.924	19.8737	0.6085	0.4335
Tallos	0.9758	94.5097	0.9698	63.6215	0.9989	2.0246	0.9073	70.4994	0.8969	0.9287
Raíces	0.9541	110.4720	0.946	24.6729	0.9955	42.3306	0.9122	1.0192	0.8784	0.9275
IAF	0.9552	25.1683	0.9195	22.5092	0.9761	20.7027	0.7868	31.4145	0.9277	0.1946

El Cuadro 4.16 indica que solo en el caso de las calibraciones para la variable peso seco total e índice de área foliar, donde el valor del coeficiente de determinación (R^2) para raíces es de 0.8942 y 0.7868 respectivamente, la predicción de la variable no presenta una tendencia similar a la observada, lo cual se puede apreciar también en las figuras presentadas en el Apéndice A, de igual manera, para el caso de las calibraciones de la variable hojas, tallos y raíces,

donde el valor del coeficiente de determinación para el índice de área foliar (IAF) es menor a 0.9, la predicción de la variable no presenta una tendencia similar a la observada. Los valores obtenidos de RMSE para las variables calibradas para el modelo SUCROS971 son ligeramente menores que los obtenidos en el modelo simple LINTUL1, sin embargo, ambos modelos predicen de una forma aceptable las variables calibradas. Las simulaciones para las variables no calibradas tienen tendencia similar a las observaciones, ya que en su mayoría presentan valores de R^2 mayores a 0.9, sin embargo, se puede apreciar que los valores de RMSE son muy elevados, lo que indica que ambos modelos sobreestiman o subestiman tales variables.

Se puede apreciar que el valor de R^2 es más elevado en las variables para las cuales se realizó la calibración, presentando valores mayores a 0.9, lo que indica que hay una tendencia similar entre predicciones y observaciones. Del mismo modo se puede observar que los valores más bajos de RMSE se presentan en las variables para las cuales se realizó la calibración, esto indica un buen ajuste entre las curvas estimadas y las observadas, lo cual se puede ver en las figuras presentadas en Apéndice B.

En el Cuadro 4.17 se muestran los valores de los parámetros para cada una de las variables calibradas. Los valores cambiaron de acuerdo a cada variable calibrada, ya que es posible que existan varias soluciones al problema de optimización en cada calibración, es decir, que sea multimodal, lo cual puede ser investigado utilizando algoritmos globales de optimización.

Cuadro 4.17. Valores de los parámetros del modelo LINTUL1 después de una calibración usando el método de mínimos cuadrados.

Variable calibrada	Parámetro			
	RGRl	LUE	K	SLA
Biomasa Total	0.0349	1.0254	0.84	0.0334
Hojas	0.0349	1.3066	0.414	0.002
Tallos	0.0389	2.2588	0.5206	0.0131
Raíces	0.007	5.5	0.95	0.002
IAF	0.0242	2.1172	0.5495	0.0086
Desviación estándar	0.0129	1.7879	0.2283	0.0129

En el Cuadro 4.17 se muestran los valores de los parámetros para cada una de las variables calibradas, existen variaciones ya que es posible que existan varias soluciones al problema de optimización en cada calibración, es decir, que sea multimodal, lo cual puede ser investigado utilizando algoritmos optimización más sofisticados.

4.3. Análisis de sensibilidad para el modelo de crecimiento limitado por agua SUCROS972.

Se realizaron simulaciones usando datos de crecimiento limitado por agua (60% de humedad de acuerdo con el cálculo de ET_0 con el método de Penman Monteith) para la variable de salida peso seco total, cambiando los valores de los parámetros de los modelos incrementándolos en distintos porcentajes (1%, 4%, 5%, 10%) para obtener y comparar los índices de sensibilidad y la función de sensibilidad de cada parámetro.

En el Cuadro 4.18 se muestran los parámetros del modelo SUCROS972 así como sus valores nominales y el índice de sensibilidad para la variable de salida peso seco total.

Cuadro 4.18. Índices de sensibilidad de los parámetros del modelo SUCROS972 para crecimiento limitado por agua.

Parámetro	Valor nominal del parámetro	Índice de Sensibilidad			
		1%	4%	5%	10%
RGRL	0.009	46.4763	47.3637	47.6642	49.2029
AMX	1.11×10^{-3}	34.4931	34.2608	34.1835	33.5005
KDF	0.6	33.9603	33.7612	33.6944	33.3572
EFF	12.5×10^{-6}	29.2384	29.066	29.0089	29.8512
SLA	0.022	21.2316	21.9769	22.2317	23.555
EES	0.002	6.2976	6.1235	6.0663	5.7877
SCP	0.2	-1.0643	-1.0733	-1.0764	-1.0924
WCAD4	0.025	0.3785	0.379	0.3791	0.3793
WCAD2	0.025	-0.1279	-0.1276	-0.1276	-0.1279
WCAD1	0.025	-0.1156	-0.1153	-0.1153	-0.1155
WCAD3	0.025	-0.0773	-0.077	-0.077	-0.0771
INTC	0.25	-2.3715×10^{-4}	1.8491×10^{-4}	2.1304×10^{-4}	2.6931×10^{-4}
MDRATE	50.0	-5.6274×10^{-4}	-1.4069×10^{-4}	-1.1255×10^{-4}	-5.6274×10^{-5}
FRTRL	0.2	-5.6274×10^{-4}	-1.4069×10^{-4}	-1.1255×10^{-4}	-5.6274×10^{-5}
LAICR	4.0	-5.6274×10^{-4}	-1.4069×10^{-4}	-1.1255×10^{-4}	-5.6274×10^{-5}
FRDR	1.0	-5.6274×10^{-4}	-1.4069×10^{-4}	-1.1255×10^{-4}	-5.6274×10^{-5}
TRANSC	9.0	-5.6274×10^{-4}	-1.4069×10^{-4}	-1.1255×10^{-4}	-5.6274×10^{-5}
WCWET1	0.35	-5.6274×10^{-4}	-1.4069×10^{-4}	-1.1255×10^{-4}	-5.6274×10^{-5}
WCWET2	0.35	-5.6274×10^{-4}	-1.4069×10^{-4}	-1.1255×10^{-4}	-5.6274×10^{-5}
WCWET3	0.35	-5.6274×10^{-4}	-1.4069×10^{-4}	-1.1255×10^{-4}	-5.6274×10^{-5}
WCWET4	0.35	-5.6274×10^{-4}	-1.4069×10^{-4}	-1.1255×10^{-4}	-5.6274×10^{-5}

El Cuadro 4.18 muestra que los parámetros más sensibles al modelo de acuerdo con el valor del índice de sensibilidad son la tasa de crecimiento relativo durante la fase de crecimiento exponencial del área foliar (RGRL), la tasa potencial de asimilación de CO₂ a saturación de luz para hojas individuales (AMX), el coeficiente de extinción de la radiación del follaje (KDF), la eficiencia del uso de luz por las hojas Individuales (EFF), el área foliar específica (SLA), el coeficiente específico de extinción del suelo (EES), el coeficiente de dispersión de las hojas para la radiación fotosintéticamente activa (SCP), los demás parámetros con índice de sensibilidad menor a este último, no se consideran sensibles al modelo, y no son tomados en cuenta para la calibración.

En las Figuras 4.22-4.28 se muestran las simulaciones para la variable peso seco total con y sin perturbación, así como la función de sensibilidad para los parámetros más sensibles al modelo SUCROS972, correspondientes a una variación de 1%.

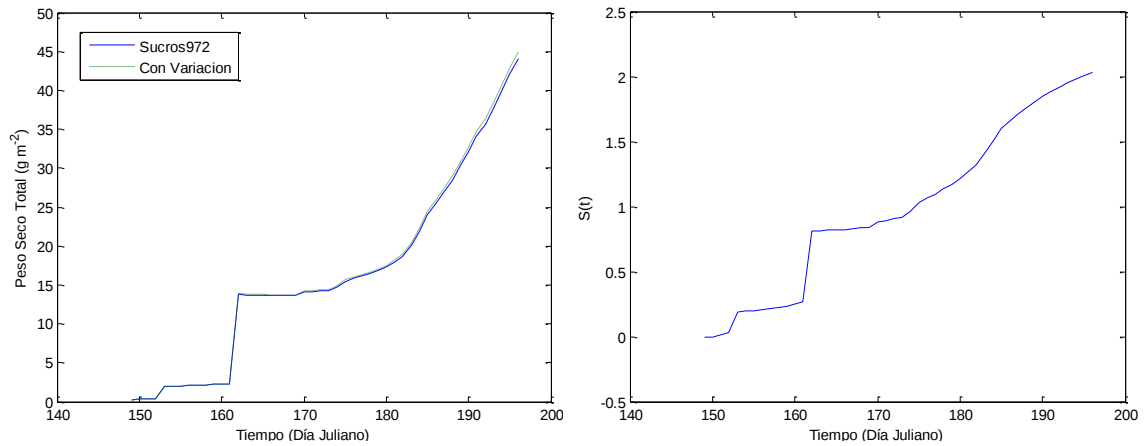


Figura 4.22. Simulaciones y función de sensibilidad para RGRL (SUCROS972).

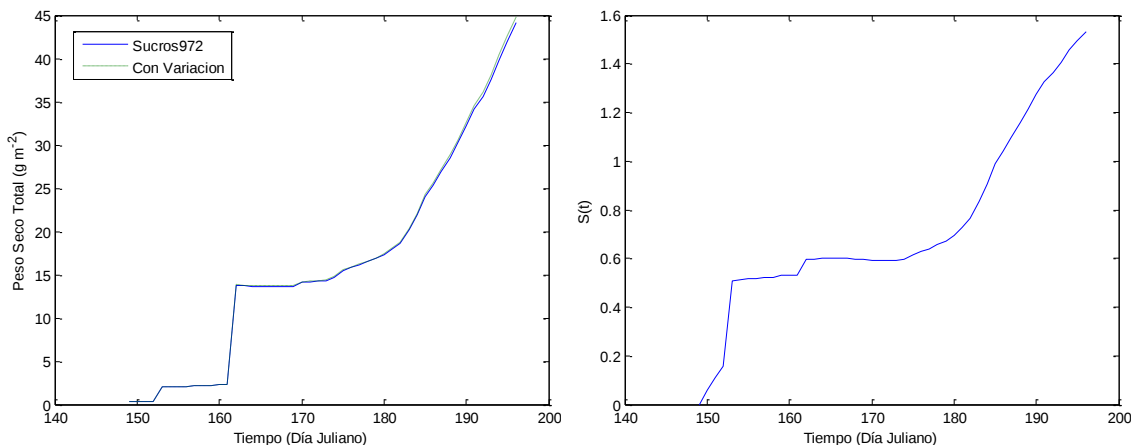


Figura 4.23. Simulaciones y función de sensibilidad para AMX (SUCROS972).

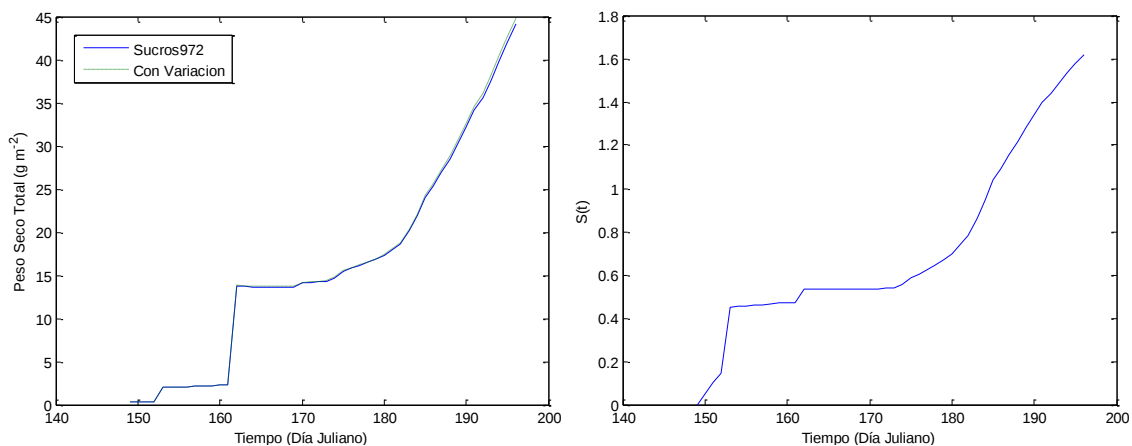


Figura 4.24. Simulaciones y función de sensibilidad para KDF (SUCROS972).

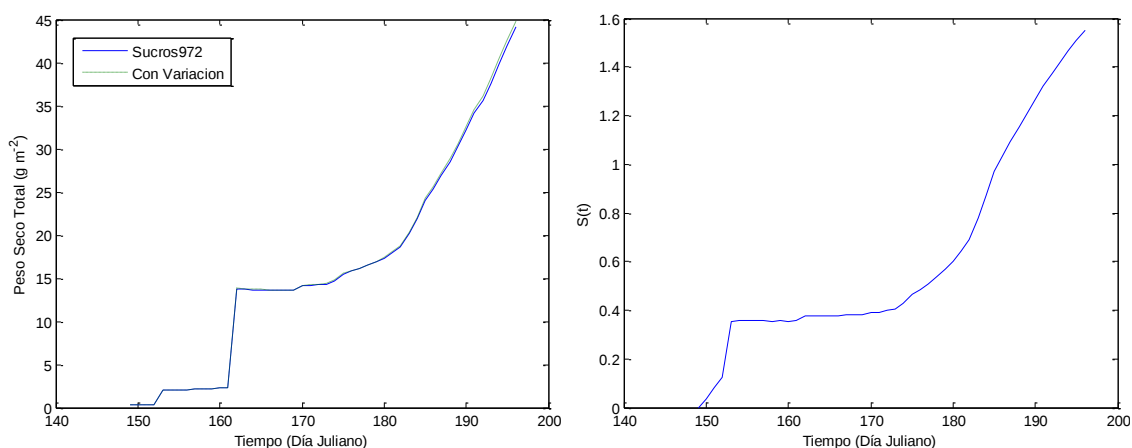


Figura 4.25. Simulaciones y función de sensibilidad para EFF (SUCROS972).

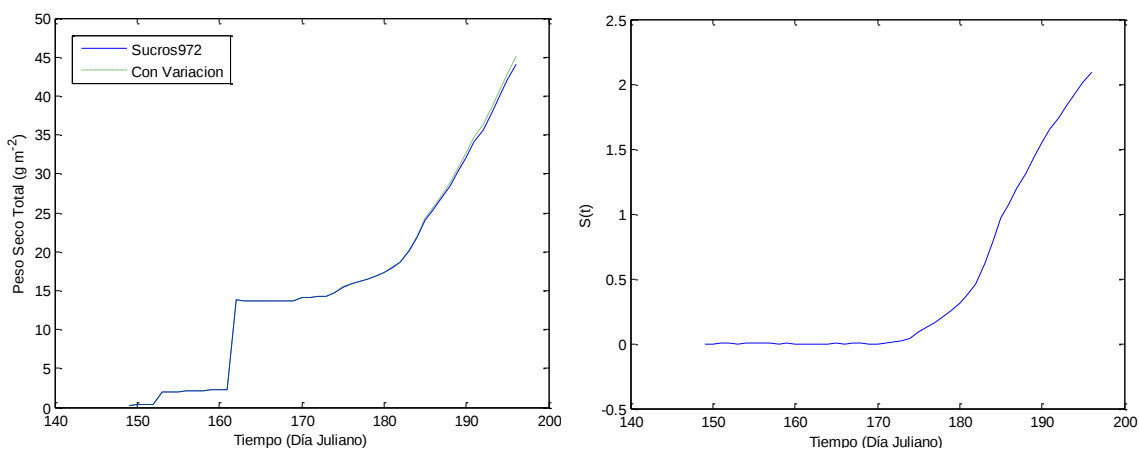


Figura 4.26. Simulaciones y función de sensibilidad para SLA (SUCROS972).

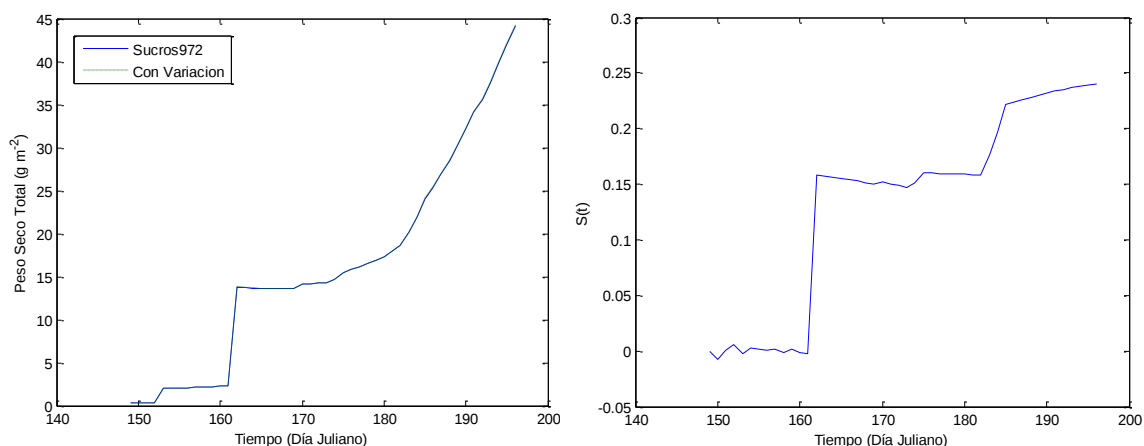


Figura 4.27. Simulaciones y función de sensibilidad para EES (SUCROS972).

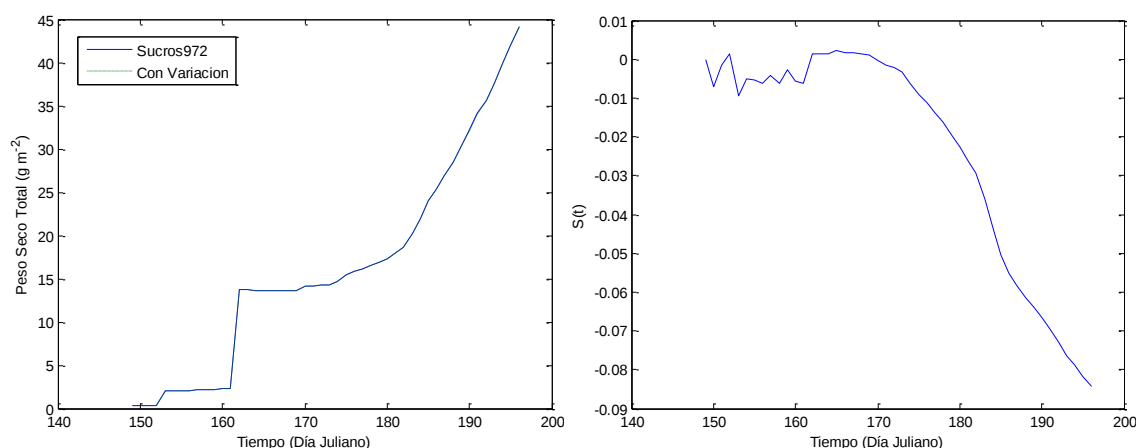


Figura 4.28. Simulaciones y función de sensibilidad para SCP (SUCROS972).

4.4. Calibración del modelo SUCROS972 para 60% de humedad.

En el Cuadro 4.19 se muestran varias ejecuciones del modelo para la variable de salida peso seco total, los límites inferior y superior para cada parámetro y sus valores correspondientes a cada ejecución. Los valores calibrados para los parámetros son los correspondientes al valor más bajo de la función a minimizar (la suma del cuadrado de los errores). En la Figura 4.29 se muestran las simulaciones con y sin calibración del crecimiento limitado por agua del cultivo de tomate de cáscara para la variable de salida peso seco total.

Cuadro 4.19. Resultados de la calibración automática del modelo de crecimiento limitado por agua para la variable peso seco total (60% de humedad).

Parámetro	Rango del parámetro	Calibración								
		1	2	3	4	5	6	7	8	9
RGRL	0.001- 0.04	0.0297	0.0297	0.0297	0.0297	0.0297	0.0297	0.0297	0.0297	0.0297
AMX	1×10^{-3} - 1.22×10^{-3}	0.0012	0.0012	0.0012	0.0012	0.0012	0.0012	0.0012	0.0012	0.0012
KDF	0.25 - 0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
EFF	12.5×10^{-7} - 2.375×10^{-5}	3.4782×10^{-6}	3.4784×10^{-6}	3.4782×10^{-6}	3.4763×10^{-6}	3.4787×10^{-6}	3.4784×10^{-6}	3.4763×10^{-6}	3.4787×10^{-6}	3.4787×10^{-6}
SLA	0.002 - 0.042	0.0419	0.0419	0.0419	0.042	0.0419	0.0419	0.042	0.0419	0.0419
EES	10^{-3} - 3×10^{-3}	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.003
SCP	0.1 - 0.3	0.2994	0.2995	0.2994	0.3	0.3	0.2995	0.3	0.3	0.3
f(x)		17.6978	17.6979	17.6978	17.6972	17.6978	17.6979	17.6972 *	17.6978	17.6978

* Valor mínimo de la función de calibración.

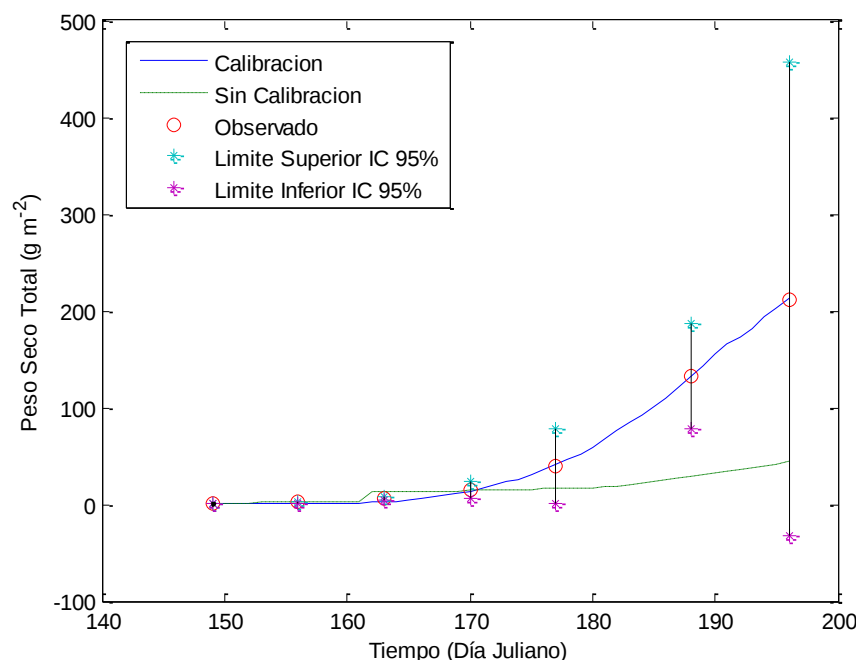


Figura 4.29. Simulación del crecimiento, sin calibración, y con calibración para la variable peso seco total (60% de humedad).

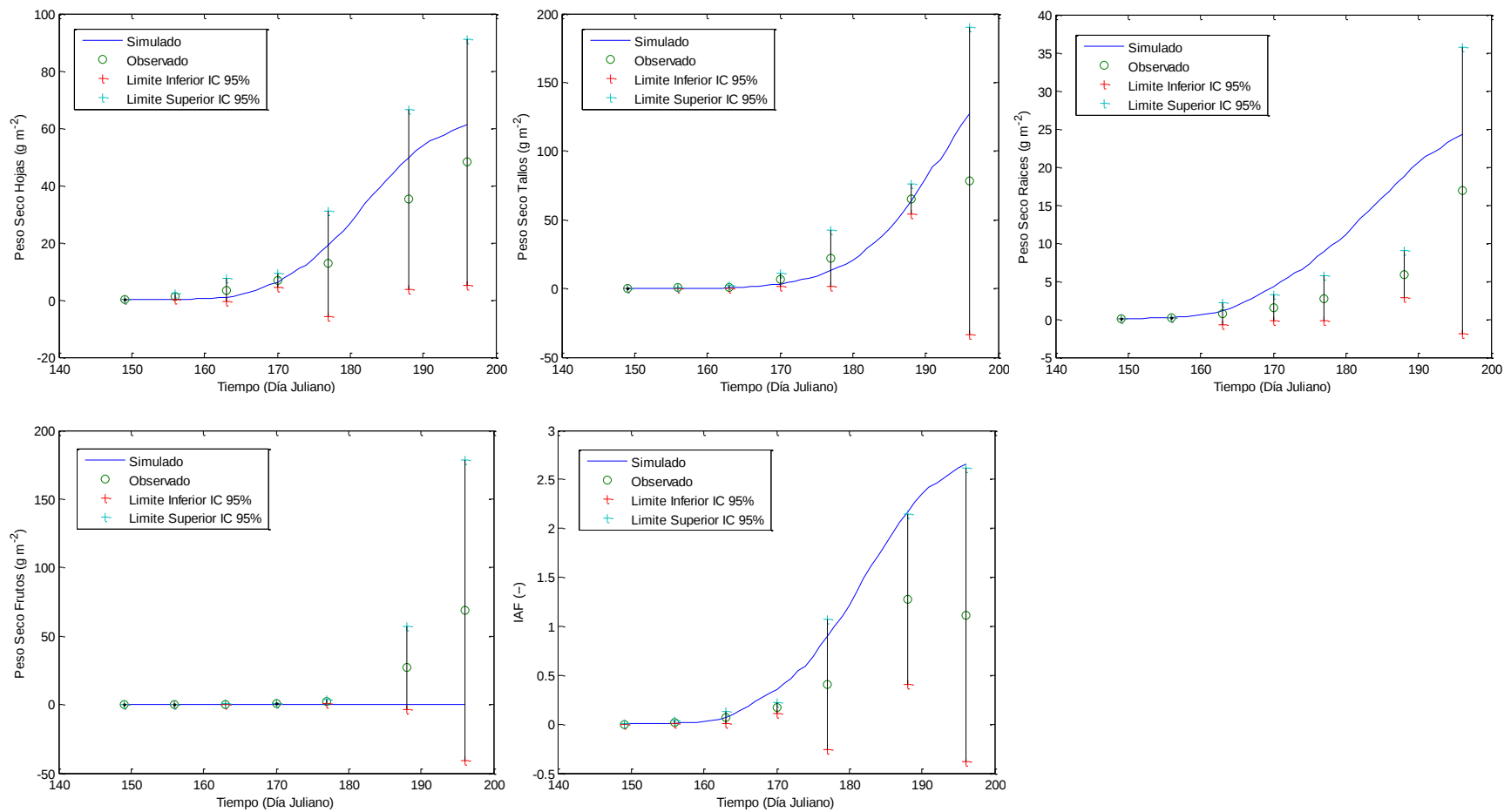


Figura 4.30. Simulaciones para las variables de estado de SUCROS972 correspondientes a 60% de humedad.

La Figura 4.29 indica que hay un buen ajuste entre la curva estimada y la observada para la variable (calibrada) peso seco total, presentan un valor para la raíz cuadrada del cuadrado medio del error $RMSE=1.59$ y un coeficiente de determinación $R^2= 0.9997$. En la figura 4.30 se muestran las simulaciones y observaciones de las variables de estado biomasa de hojas ($R^2=0.9902$; $RMSE=7.8310$), tallos ($R^2=0.9143$; $RMSE=18.7862$), raíces ($R^2=0.839$; $RMSE=6.1422$), frutos, e índice de área foliar ($R^2=0.9539$; $RMSE=0.7015$).

Todas las variables presentan una buena tendencia en la curva estimada a excepción de la biomasa de frutos donde el modelo no es capaz de predecir de manera apropiada la variable, únicamente para el caso de la biomasa de raíces se presenta un valor menor a 0.9 para R^2 . En la Figura 4.30 se puede apreciar que el modelo sobreestima la biomasa de hojas, raíces y el índice de área foliar.

4.5. Calibración del modelo SUCROS972 para 80% de humedad.

Se realizó una calibración automática para los parámetros que hacen más sensible al modelo, usando datos de 80% de la humedad óptima, se obtuvieron los valores que a continuación se presentan en el Cuadro 4.20. En la Figura 4.31 se muestran las simulaciones con y sin calibración del crecimiento limitado por agua del cultivo de tomate de cáscara para la variable peso seco total ($R^2=0.9629$; $RMSE=15.2818$).

Cuadro 4.20. Resultados de la calibración automática del modelo de crecimiento limitado por agua para la variable peso seco total (80% de humedad).

Parámetro	Rango del parámetro	Calibración								
		1	2	3	4	5	6	7	8	9
RGRL	0.001- 0.04	0.0155	0.0155	0.0155	0.0155	0.0155	0.0155	0.0155	0.0155	0.0155
AMX	1×10^{-3} - 1.22×10^{-3}	0.0012	0.0012	0.0012	0.0012	0.0012	0.0012	0.0012	0.0012	0.0012
KDF	0.25 - 0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
EFF	12.5×10^{-7} - 2.375×10^{-5}	2.3750×10^{-5}	2.3750×10^{-5}	2.3750×10^{-5}	2.3750×10^{-5}	2.3750×10^{-5}	2.3750×10^{-5}	2.3750×10^{-5}	2.3750×10^{-5}	2.3750×10^{-5}
SLA	0.002 - 0.042	0.0196	0.0196	0.0196	0.0196	0.0196	0.0196	0.0196	0.0196	0.0196
EES	10^{-3} - 3×10^{-3}	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
SCP	0.1 - 0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3
CME		1634.73	1634.73	1634.73	1634.73	1634.73	1634.73	1634.73	1634.73	1634.73 *

* Valor mínimo de la función de calibración.

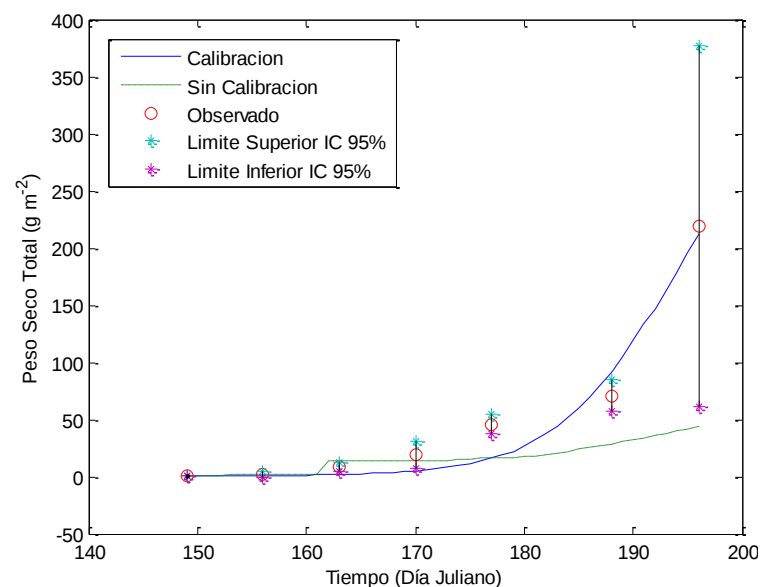


Figura 4.31. Simulación del crecimiento, sin calibración, y con calibración para la variable peso seco total (80% de humedad).

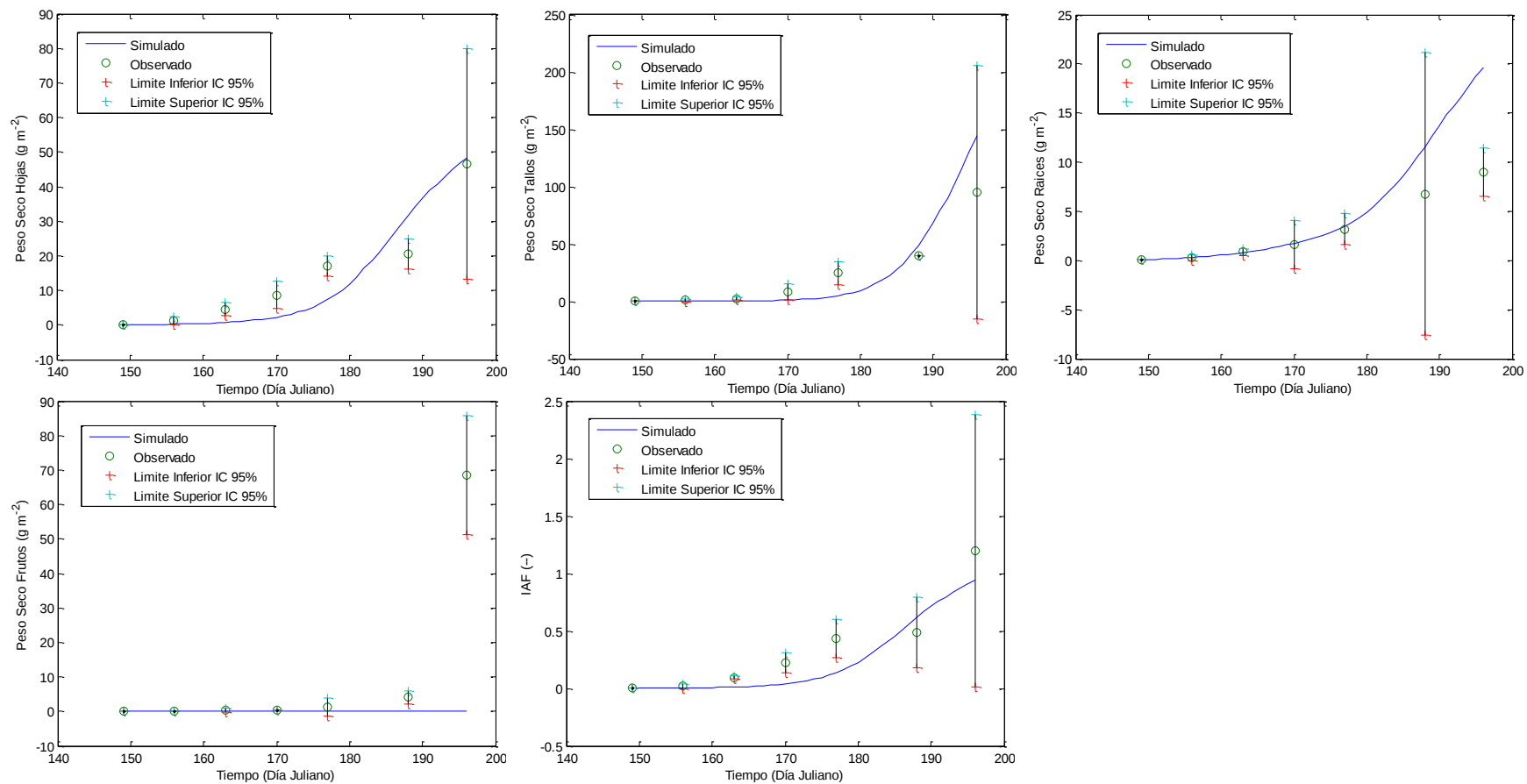


Figura 4.32. Simulaciones para las variables de estado de SUCROS972 correspondientes a 80% de humedad.

Para la calibración correspondiente a un 60% de humedad se tiene un buen ajuste de la función de calibración con un valor de la función de calibración de 17.6972, para el caso de 80% se tiene un ajuste más pobre que corresponde a un valor de 1634.73. En ambos casos el modelo es capaz de predecir de manera apropiada la variable peso seco total, sin embargo, se puede observar en la Figura 4.31 que el valor observado correspondiente al día juliano 188 es muy bajo en relación con el crecimiento normal del cultivo, afectando de manera significativa el ajuste de la curva de calibración.

En la Figura 4.32 se presentan las simulaciones para las variables de estado biomasa de hojas ($R^2= 0.8873$; RMSE= 6.3099), tallos ($R^2= 0.9545$; RMSE= 20.5083), raíces ($R^2= 0.9626$; RMSE= 4.4309), frutos, e índice de área foliar ($R^2= 0.8678$; RMSE= 0.1739). Aunque todas las variables presentan valores de R^2 elevados, en general, las estimaciones para ambos niveles de humedad son pobres. Se puede ver que de la misma manera que para el caso de la calibración para 60% de humedad, el modelo presenta dificultades para estimar la biomasa de frutos.

Capítulo 5. CONCLUSIONES

Los resultados de las simulaciones mostraron que el modelo de crecimiento potencial SUCROS971 mostró un buen ajuste entre las predicciones y los datos observados en campo para la variable de salida peso seco total, para las variables de estado biomasa de hojas, biomasa de tallos y biomasa de frutos, sin embargo, el ajuste fue pobre en las variable biomasa de raíces e índice de área foliar. El modelo simple LINTUL1 presentó un ajuste pobre entre las simulaciones y los datos observados y el modelo para crecimiento limitado por agua SUCROS972 al igual que LINTUL1 tuvo dificultades para estimar la variable de salida peso seco total.

Los resultados de las calibraciones de los modelos para crecimiento potencial mostraron que el modelo SUCROS971 es capaz de predecir de manera aceptable la variable de salida peso seco total, y las variables de estado biomasa de hojas, biomasa de tallos, biomasa de raíces, biomasa de frutos e índice de área foliar. El modelo LINTUL1 predice de manera aceptable todas las variables a excepción de la biomasa de frutos.

Para cada variable calibrada se hicieron simulaciones en ambos modelos de todas las demás variables y se observó que el modelo SUCROS971 estima de manera aceptable el comportamiento de las variables, a excepción de la biomasa de raíces e índice de área foliar. El modelo LINTUL1 presenta

dificultades, ya que en algunas ocasiones sobreestima y en otras subestima el comportamiento de las variables en la mayoría de las simulaciones.

Los resultados de la calibración del modelo SUCROS972 para la variable de salida peso seco total para el 60% de humedad, indican que el modelo es capaz de predecir de manera aceptable el comportamiento de la variable. Los resultados de la calibración del modelo SUCROS972 para la variable de salida peso seco total para el 80% de humedad indican que el modelo es capaz de predecir el comportamiento de la variable de manera aceptable pero más pobre que para el nivel de humedad de 60%, debido a un posible error de medición al momento del penúltimo muestreo en campo, ya que el valor observado es muy pequeño de acuerdo con el crecimiento del cultivo.

En general, los modelos para crecimiento potencial SUCROS971 y LINTUL1, y el modelo para crecimiento limitado por agua SUCROS972 son capaces de estimar el comportamiento del cultivo de tomate de cascara en condiciones de acolchado plástico para los niveles de humedad 60%, 80% y 100%, sin embargo se requiere realizar experimentos que permitan incorporar información más detallada de la distribución de la biomasa de frutos, para lograr un mejor ajuste entre las observaciones y las estimaciones de los modelos.

Se recomienda utilizar en trabajos futuros métodos globales de optimización para mejorar aún más las predicciones de los modelos, y llevar a cabo análisis de sensibilidad globales que permitan una exploración más a detalle de la incertidumbre y las sensibilidades de los parámetros.

LITERATURA CITADA

- Aguilar, Z. R. y Méndez J. Moisés A. 2004. Descripción Varietal de Tomate de Cáscara. Tesis Profesional. Departamento de Fitotecnia de la Universidad Autónoma Chapingo. Chapingo, México.
- Allen, G. R.; Pereira S. L.; Raes D; Smith M. 1998. Crop Evapotranspiration. FAO Irrigation and Drainage Paper No. 56. FAO, Rome, Italy. 300 p.
- Bindi, M. 2003. Instrumentos para el monitoreo del impacto ambiental sobre la producción agrícola. Modelos para cultivos. Università degli Studi di Firenze.
- Boote, K. J.; Jones J. W., and Pickering, N. B. 1996. Potential uses and limitations of crop models. *Agronomy Journal*. 88:704-716.
- Bouman, B. A. M., van Keulen, H., van Laar H. H. and Rabbinge, R. 1996. The 'School of the Wit' Crop Growth Simulation Models: a pedigree and historical overview. *Agricultural Systems*, Vol. 52, Nos 2/3, pp. 171-198.
- Brisson, N.; Gary, C.; Justes, E.; Roche, R.; Mary, B.; Ripoche, D.; Zimmer, D.; Sierra, J.; Bertuzzi, P.; Burger, P.; Bussière, F.; Cabidoche, Y. M.; Cellier, P.; Debaeke, P.; Gaudillère, J. P.; Hénault, C.; Maraux, F.; Seguin, B.; Sinoquet, H. 2003. An overview of the crop model STICS. *European Journal of Agronomy*. 18. 309-332.
- Brun, R., Reichert, P., Kunsch, H.R., 2001. Practical identifiability of large environmental simulation models. *Water Resources Research* 37, 1015–1030.

- Butterbach-Bahl, K., Kesik, M., Miehle, P., Papen, H., Li, C., 2004. Quantifying the regional source strength of N-trace gases across agricultural and forest ecosystems with process based models. *Plant and Soil* 260, 311–329.
- Cariboni, J., Gatelli, D., Liska, R., Saltelli, A. The role of Sensitivity Analysis in ecological modeling. *Ecological modelling* 203 (2007) 167–182.
- Chonggang Xu, George Zdzislaw Gertner. 2007. A general first-order global sensitivity analysis method. *Reliability Engineering and System Safety*. Volume 93, Issue 7, Pages 1060-1071.
- Cooman, A.; Schrevens, E. 2007. Sensitivity of the Tomgro model to solar radiation intensity, air temperature and carbon dioxide concentration. *Biosystems Engineering*. 96 (2), 249–255.
- De Wit, C.T. 1982. Simulation of living systems. In *Simulation of plant growth and crop production*. Eds. F.W.T. Penning de Vries and H.H. van Laar. *Simulation Monographs*. PUDOC. Wageningen. The Netherlands. pp. 3-8.
- Escuela de Graduados C. T. de Wit de Ecologia de la Produccion de la Universidad de Wageningen, Holanda, 1997. A simple general crop growth model for optimal growing conditions (example: spring wheat). P.O. Box 430, 6700 AK Wageningen.

- Gabrielle, B., Laville, P., He´ nault, C., Nicoullaud, B., Germon, J.C., 2006a. Simulation of nitrous oxide emissions from wheat-cropped soils using CERES. *Nutrient Cycling in Agroecosystems* 74, 133–146.
- Goudriaan, J. 1988. The bare bones of leaf-angle distribution in radiation models for canopy photosynthesis and energy exchange. *Agric. Forest Meteorol.* 43: 155-169.
- Goudriaan, J. and J. L. Monteith. 1990. Mathematical function for crop growth based on light interception and leaf area expansion. *Ann. Bot.* 66: 695-701.
- Goudriaan, J., and H. H. Van Laar. 1994. Modelling potential crop growth processes. Textbook with exercises. Kluwer Academic Publishers. Dordrecht. 238 pp.
- Jones, J. W.; Hoogenboom, G.; Porter, C. H.; Boote, K. J.; Batchelor, W. D.; Hunt, L. A.; Wilkens, P. W.; Singh, U.; Gijsman, A. J.; Ritchie, J. T. 2003. The DSSAT cropping system model. *European Journal of Agronomy.* 18. 235-265.
- Keating, B. A.; Carberry, P. S.; Hammer, G. L.; Probert, M. E.; Robertson, M. J.; Holzworth, D.; Huth, N. I.; Hargreaves, J. N. G.; Meinke, H.; Hochman, Z.; McLean, G.; Verburg, K.; Snow, V.; Dimes, J. P.; Silburn, M.; Wang, E.; Brown, S.; Bristow, K. L.; Asseng, S.; Chapman, S.; McCown, R. L.; Freebairn, D. M.; Smith, C. J. 2003. An overview of APSIM, a model

designed for farming systems simulation. *European Journal of Agronomy*.
18. 267-288.

Lamboni, M., Makowski, D., Lehuger, S., Gabrielle, B., Monod, H. 2009.
Multivariate global sensitivity analysis for dynamic crop models. *Field
Crops Research* 113 (2009) 312–320.

Lehuger, S., Gabrielle, B., VanOijen, M., Makowski, D., Germon, J.C., Morvan,
T., Henault, C., 2009. Bayesian calibration of the nitrous oxide emission
module of an agro-ecosystem model. *Agric. Ecosyst. Environ*,
doi:10.1016/j.agee.2009. 04.022.

López, C. I. L., Ramírez, A. A., Rojano, A. A. 2004. Analisis de sensibilidad de
un modelo dinamico de crecimiento para lechugas (*Lactuca Sativa* L.)
cultivadas en invernadero. *Agrociencia*. Vol. 38. Numero 006. pp. 613-
624.

López, C. I. L., Ramírez, A. A., Rojano, A. A. 2005. Modelos matemáticos de
hortalizas en invernadero trascendiendo la contemplación de la dinámica
de cultivos. *Revista Chapingo. Serie Horticultura*. Vol. 11. Número 002.
pp 257-267.

López, C. I. L., Rojano, A. A., Ramírez, A. A., y Bonilla, E. M. s/f. Growth and
Development Crops Mathematical Models: Concepts and Methodology.
34 pp.

López, C. I. L., Arteaga, R. R., Robles, B, C. 2008. Predicción del crecimiento
potencial de tomate de cáscara mediante los modelos LINTUL y

SUCROS. En La ingeniería agrícola - motor del desarrollo de la agricultura mexicana. XXIII congreso nacional de ingeniería agrícola. Universidad Autónoma Chapingo.

López, L., R. 2008. Necesidades hídricas e índice de estrés hídrico en tomate de cáscara (*Physalis ixocarpa* Brot.). Tesis de Doctorado. Universidad Autónoma Chapingo.

López, L., R; Arteaga R., R; Vázquez P., M. A.; López C., I.; Sánchez C., I. 2009. Índice de estrés hídrico como un indicador del momento de riego en cultivos agrícolas. Agricultura Técnica en México, Instituto Nacional de Investigaciones Forestales, Agrícolas y Pecuarias Texcoco, México. Vol. 35, Núm. 1, pp. 92-106.

López, L., R.; Arteaga R., R.; Vázquez P., M. A.; López C., I. L.; Sánchez C., I.; Ruiz G., A. 2009. Índice de estrés hídrico del cultivo de tomate de cáscara (*Physalis ixocarpa* Brot.) Revista Chapingo. Serie horticultura, Universidad Autónoma Chapingo, México. Vol. 15, Núm. 3, pp. 259-267.

López, L., R.; Arteaga R., R.; Vázquez P., M. A.; López C., I. L.; Sánchez C., I. 2009. Producción de tomate de cáscara (*Physalis ixocarpa* Brot.) basado en laminas de riego y acolchado plástico. Revista Chapingo. Serie Horticultura, Universidad Autónoma Chapingo. 15(1): 83-89.

Makowski, D., Hillier, J., Wallach, D., Andrieu, B., Jeuffroy, M.-H., 2006a. Parameter estimation for crop models. In: Wallach, D., Makowski, D.,

- Jones, J. (Eds.), Working with Dynamic Crop Models. Elsevier, Amsterdam, pp. 101–150.
- Moreno, M., M.; Peña, L., A.; Sahagún, C., J.; Rodríguez, P., J. E.; Mora, A., R. 2002. Varianza aditiva, heredabilidad y correlaciones en la variedad M1-Fitotecnia de tomate de cáscara (*Physalis ixocarpa*, Brot.). Revista Fitotecnia Mexicana. 25(3): 231-237.
- Morris, M. D. 1991. Factorial sampling plans for preliminary computational experiments. *Techometrics*. Vol. 33, No. 2.
- Metselaar, K., 1999. Auditing predictive models: a case study in crop growth. PhD-thesis, Wageningen Agricultural University, Wageningen, The Netherlands, pp. 265.
- Monteith, J. L. 1990. Conservative behaviour in response of crops to water and light. In: *Theoretical Production Ecology: Reflections and Prospects*. R. Rabbinge, J. Goudriaan, H. van Keulen, F. W. T. Penning de Vries and H. van Laar (Eds.). Simulation Monographs 34. Wageningen, The Netherlands: PUDOC, pp. 3-16.
- Peart, Robert M.; R. Bruce Curry. 1998. Agricultural systems modeling and simulation. Marcel Dekker, Inc. The University of Florida, Gainesville, Florida.
- Penning de Vries, F.W.T.; Jansen D.M.; ten Berge, H.F.M.; Bakema A. 1989. Simulation of ecophysiological processes of growth in several annual crops. Pudoc Wageningen, pp. 271.

- Peña, L., A.; Molina, G., J. D.; Márquez, S., F.; Sahagún, C., J.; Ortiz, J., C.; Cervantes, S., T. 2002. Respuestas estimadas y observadas de tres métodos de selección en tomate de cascara (*Physalis ixocarpa* Brot.). *Revista Fitotecnia Mexicana* 25(2): 171-178.
- Peña, L. A.; Santiaguillo, H. J. F. 1998. Variabilidad genética de tomate de cáscara en México. Boletín Técnico No. 2. Programa nacional de investigación en olericultura. Departamento de Fitotecnia. Universidad Autónoma Chapingo. Chapingo, México. 26 p.
- Peña, L., A.; Molina, G., J. D.; Sahagún, C., J.; Ortiz, C., J.; Márquez, S., F.; Cervantes, S., T.; Santiaguillo, H., J. F. 2008. Parámetros genéticos en la variedad CHF1 Chapingo de tomate de cáscara (*Physalis ixocarpa*, Brot.). *Revista Chapingo, Serie Horticultura*. 14(1): 5-11.
- Peña, L., A. 2001. Situación actual y perspectivas de la producción y mejoramiento genético de tomate de cáscara (*Physalis ixocarpa*, Brot.) en México.
- Poluektov, R. A., Topaj A. G. 2001. Crop modeling: nostalgia about present or reminiscence about future. *Agronomy Journal* 93: 653-659.
- Ruiz, G. A. 2009. Modelos para la simulación y control del clima en un invernadero con ventilación natural. Tesis de Maestría. Universidad Autónoma Chapingo.
- Saltelli, A.; Chan, K.; Scott E. M. 2000. Sensitivity analysis. Wiley series in probability and statistics.

- Saltelli, A., Tarantola, S., Campolongo, F. 2000. Sensitivity analysis as an ingredient of modeling. *Statistical Science*. Vol. 15, No. 4, 377-395.
- Saltelli, A. 2002. Sensitivity analysis for importance assessment. *Risk Analysis*. Vol. 22, No. 3.
- SARH. 2001. Anuario Estadístico de la Producción Agrícola de los Estados Unidos Mexicanos Tomo I. Mexico, D.F. pp. 65.
- Sinclair, T. R. 1990. Nitrogen influence on the physiology of crop yield. In: *Theoretical Production Ecology: Reflections and Prospects*. R. Rabbinge, J. Goudriaan, H. van Keulen, F. W. T. Penning de Vries, and H. H. van Laar (Eds.). *Simulation Monographs 34*. Wageningen, The Netherlands: PUDOC, pp.41-55.
- Spitters, C. J. T., H. Van Keulen and D. W. G. Van Kraalingen, 1989. A simple and universal crop growth Simulator SUCROS87. In: *Simulation and Systems Management in Crop Protection*. Eds. R. Rabbinge, S. A. Ward and H. H. Van Laar. *Simulation Monographs 32*, Pudoc, Wageningen, pp. 147-181.
- Spitters, C. K. T. 1990. Crop growth models: their usefulness and limitations. Centre for Agrobiological Research (CABO). P. O. Box 14, 6700 AA Wageningen, The Netherlands. *Acta Horticulturae* 267.
- Stöckle, C. O.; Donatelli, M.; Nelson, R. 2003. CropSyst, a cropping systems simulation model. *European Journal of Agronomy*. 18. 289-307.

- Thornley J.H.M., Johnson I. R. 2000. Plant and crop modeling. A mathematical approach to plant and crop physiology, The Blackburn Press.
- Tremblay, M., Wallach, D., 2004. Comparison of parameter estimation methods for crop models. *Agronomie* 24, 351–365.
- Valtierra, P., E.; Ramos, S., A. 2003. Programa estratégico de necesidades de Investigación y transferencia de tecnología de la cadena productiva de tomate verde en el Estado de Puebla. Fundación PRODUCE Puebla A.C.
- Van Ittersum, M.K.; Leffelaar P.A.; van Keulen H.; Kropff M.J.; Bastiaans L.; Goudriaan, J. 2003. On approaches and applications of the Wageningen crop models. *European Journal of Agronomy* 18: 201-234.
- Van Laar H.H., J. Goudriaan, H van Keulen. 1997. SUCROS97: simulation of crop growth for potential and water-limited production situations. AD-DLO. Wageningen, The Netherlands.
- Verdoodt, A., Van Ranst, E., Ye, L. 2004. Daily Simulation of Potential Dry Matter Production of Annual Field Crops in Tropical Environments. *Agronomy Journal*. 96:1739–1753.
- Wallach, D., Goffinet, B., Bergez, J. E., Debaeke, P., Leenhardt, D., Aubertot, J. N. 2001. Parameter estimation for crop models: a new approach and application to a corn model. *Agronomy Journal*. 93:757–766.
- Wallach, D., Makowsky, D., Jones, J.W. 2006. Working with dynamic crop models. Evaluation, analysis, parametrization and applications. Elsevier, pp. 447.

Xionyou, Yin.; H. H. van Laar. 2005. Crop Systems Dynamics. Crop and Weed Ecology Group, Wageningen University and Research Centre, P. O. Box 430, 6700 AK Wageningen, The Netherlands.

Apéndice A

Simulaciones para las variables de estado del modelo de crecimiento potencial SUCROS971

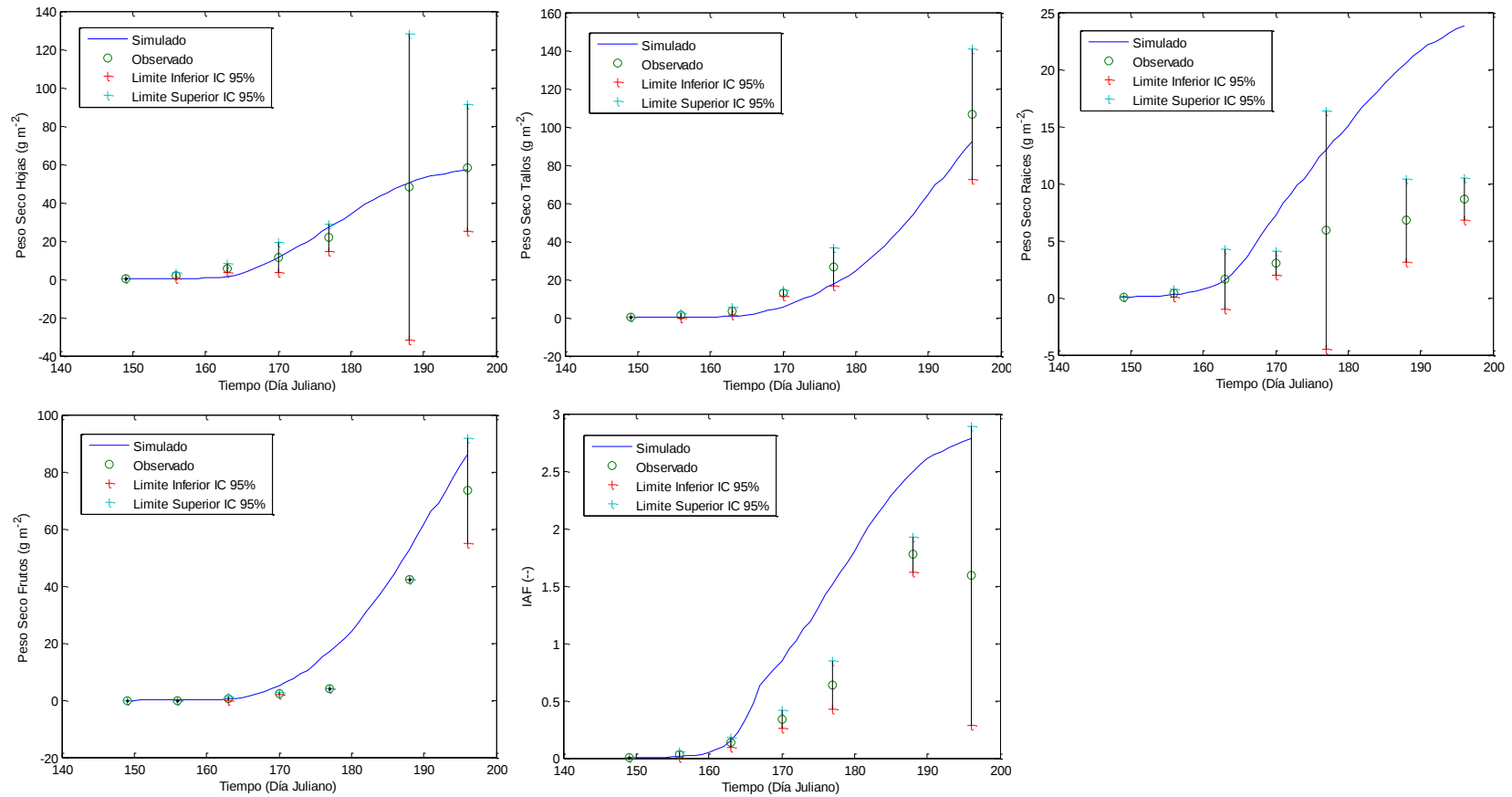


Figura 1.A. Simulaciones de las variables de estado correspondientes a la calibración de la variable peso seco total.

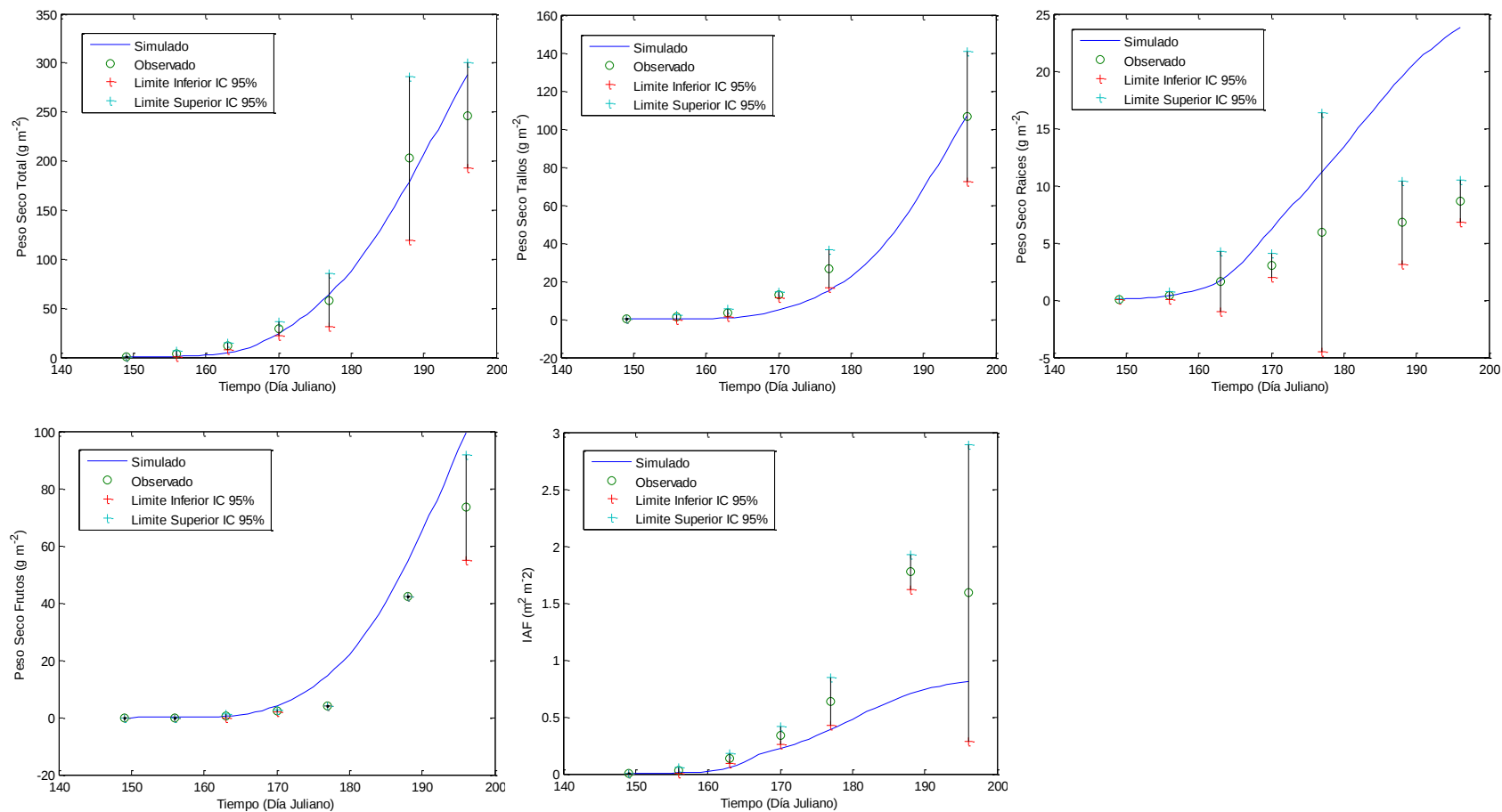


Figura 2.A. Simulaciones de las variables de estado correspondientes a la calibración de la variable peso seco de hojas.

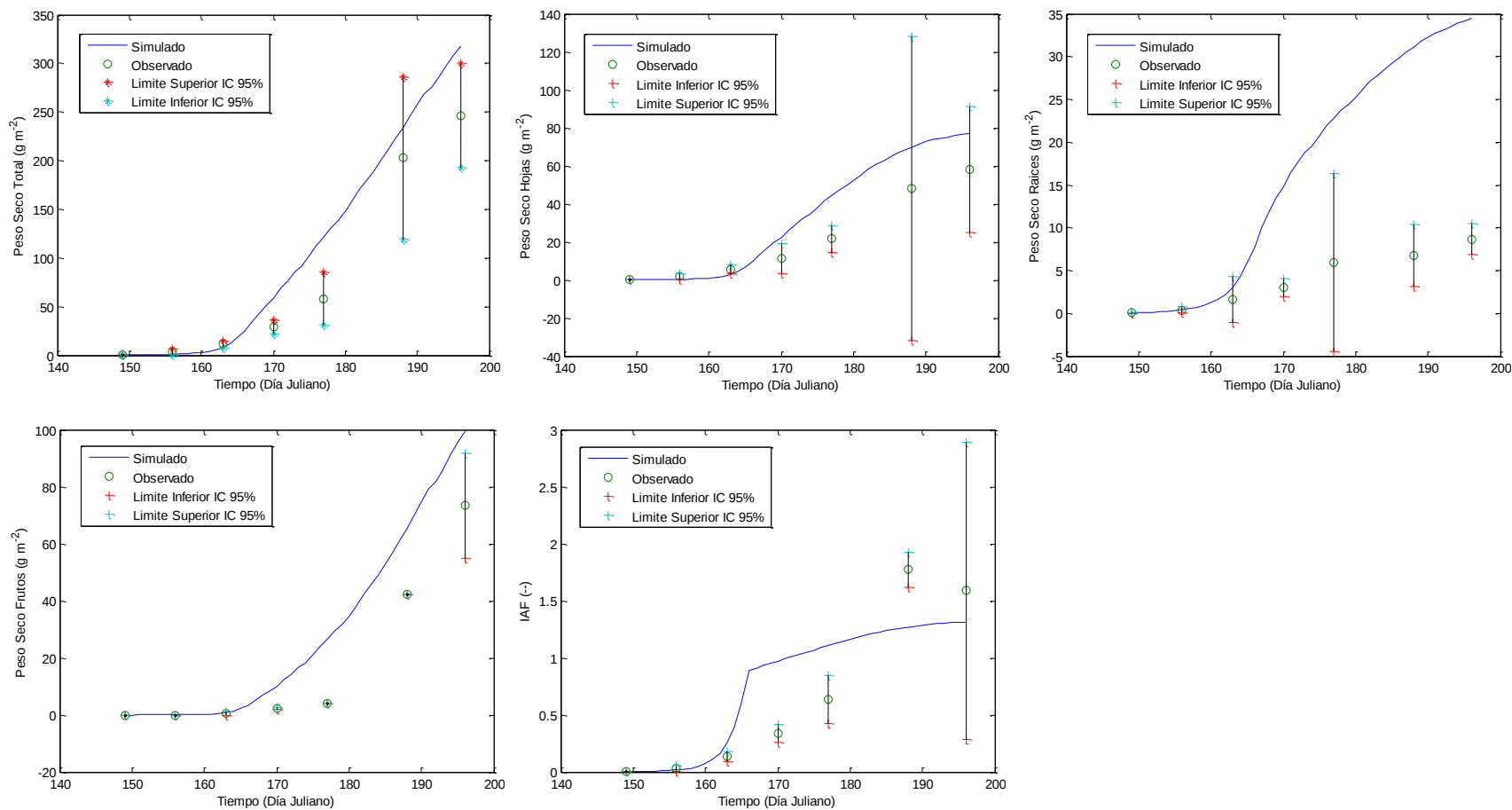


Figura 3.A. Simulaciones de las variables de estado correspondientes a la calibración de la variable peso seco de tallos.

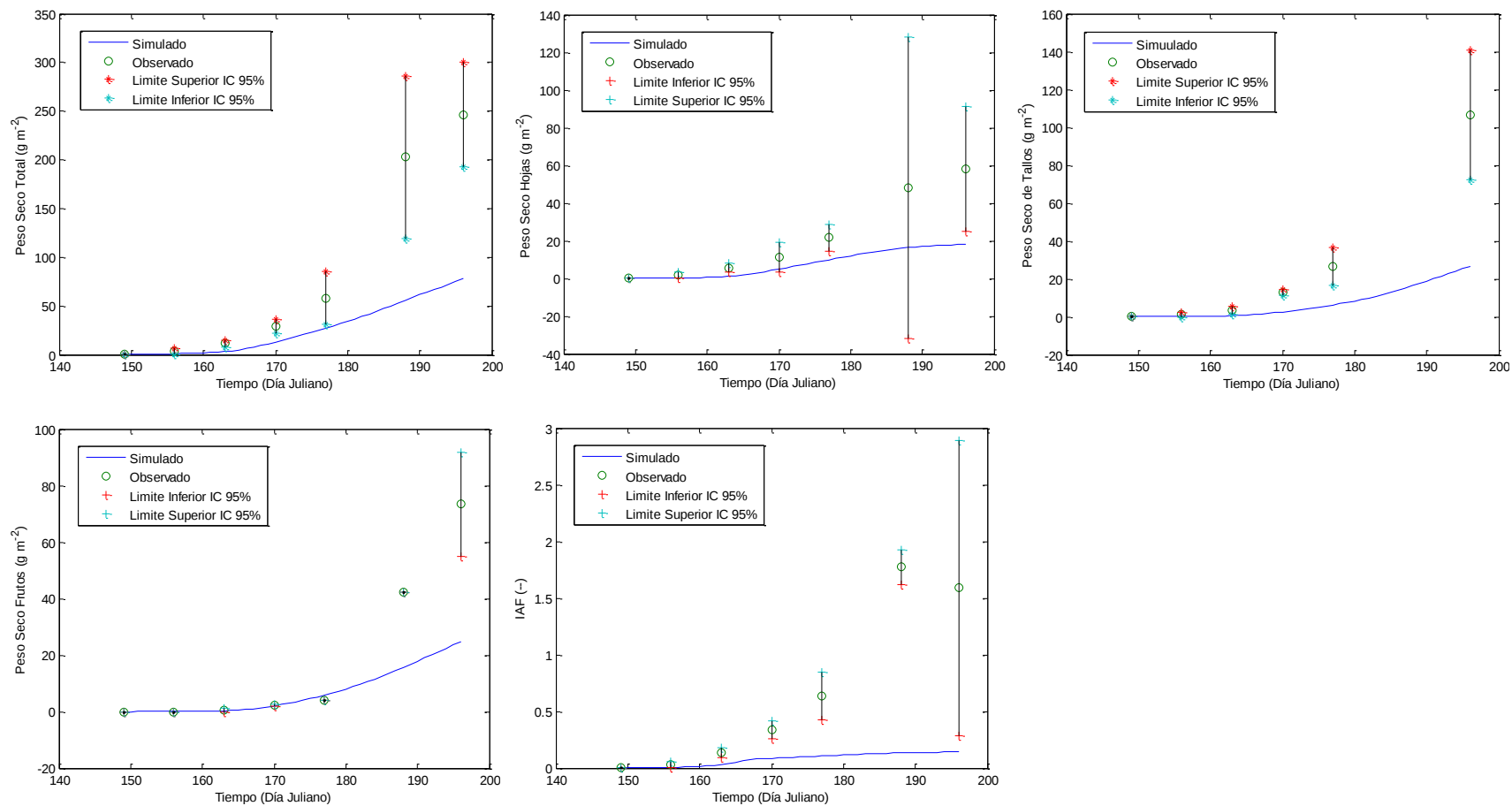


Figura 4.A. Simulaciones de las variables de estado correspondientes a la calibración de la variable peso seco de raíces.

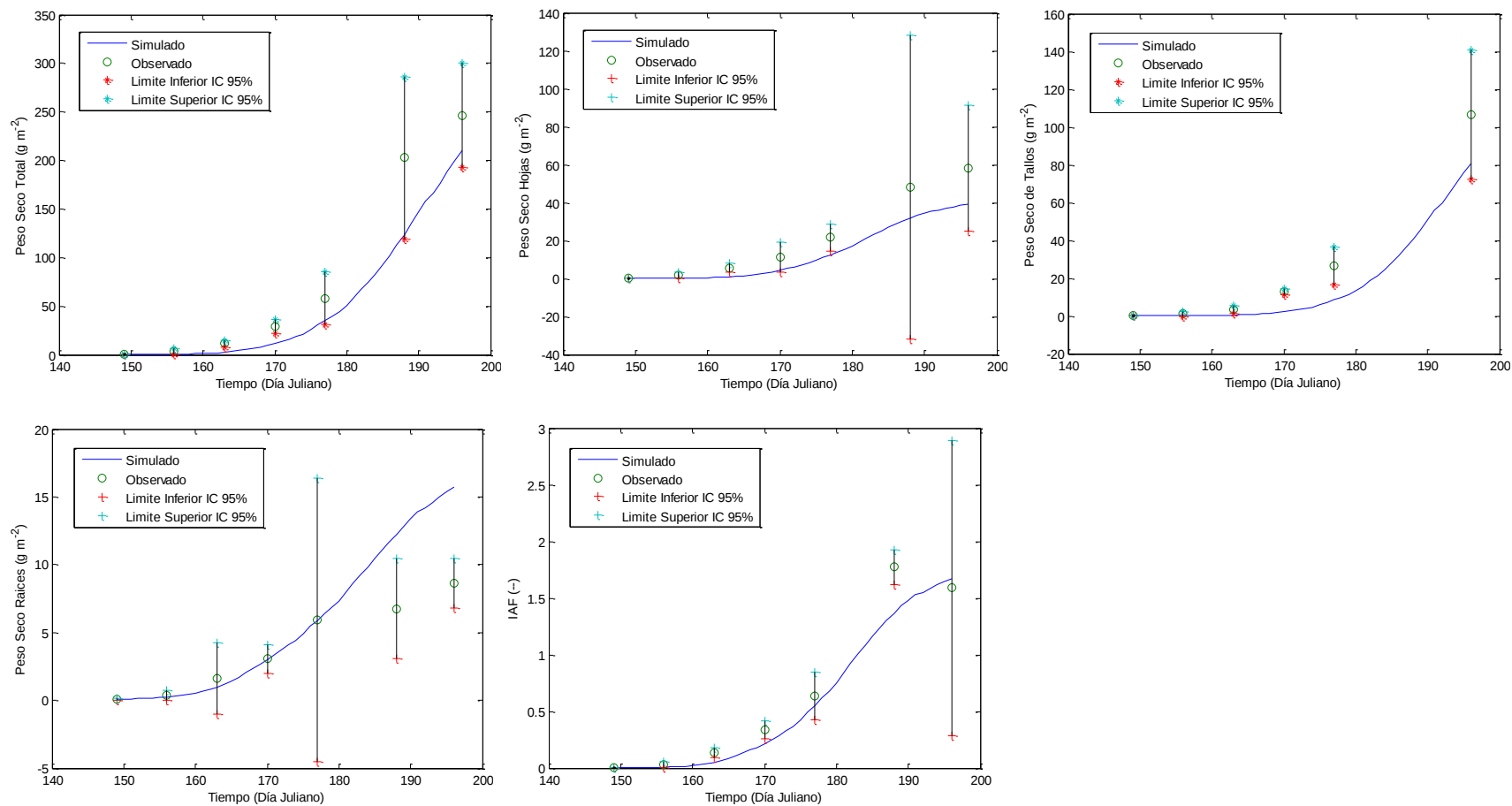


Figura 5.A. Simulaciones de las variables de estado correspondientes a la calibración de la variable peso seco de frutos.

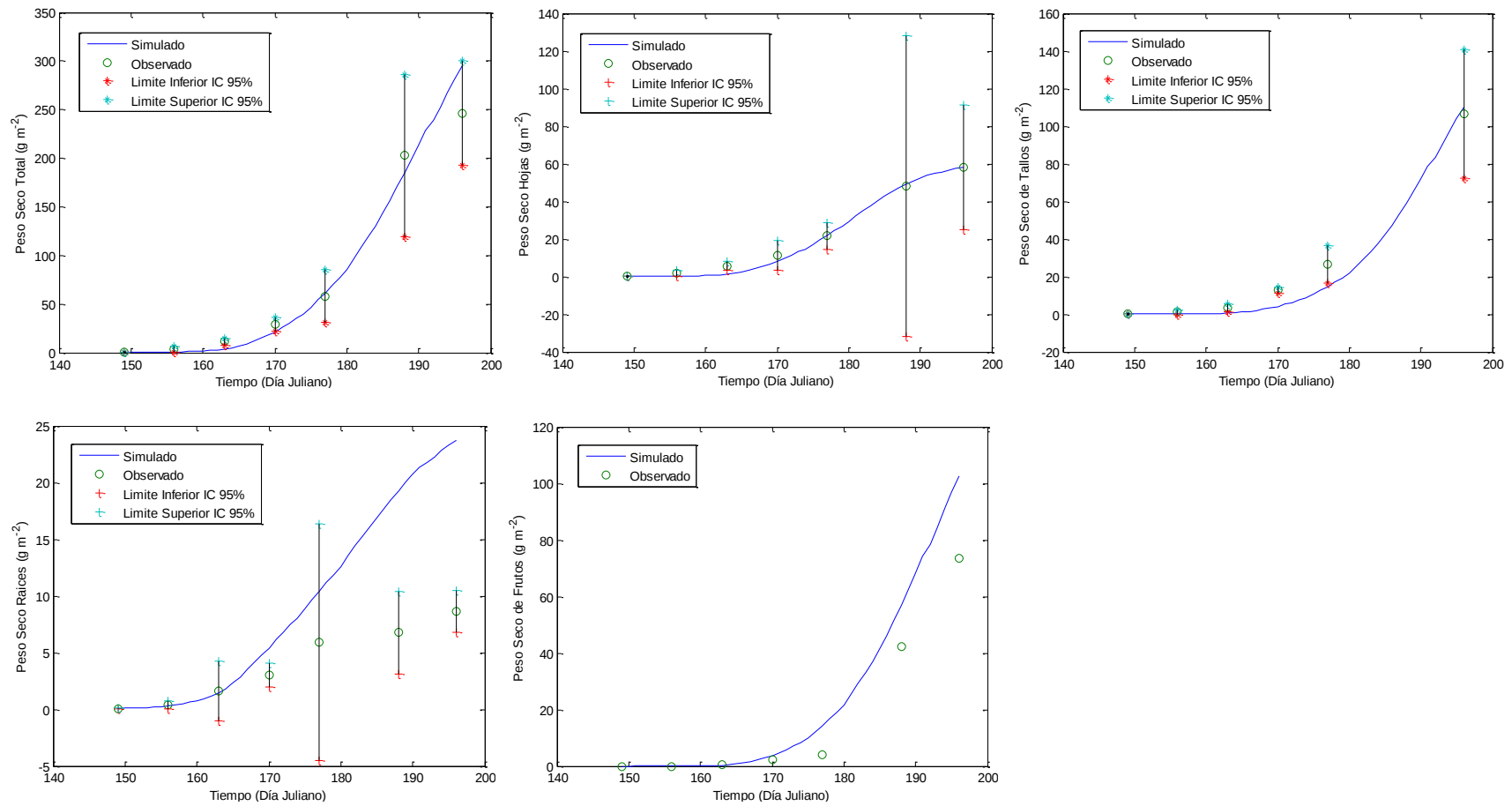


Figura 6.A. Simulaciones de las variables de estado correspondientes a la calibración de la variable índice de área foliar.

Apéndice B

Simulaciones para las variables de estado del modelo de crecimiento potencial LINTUL1

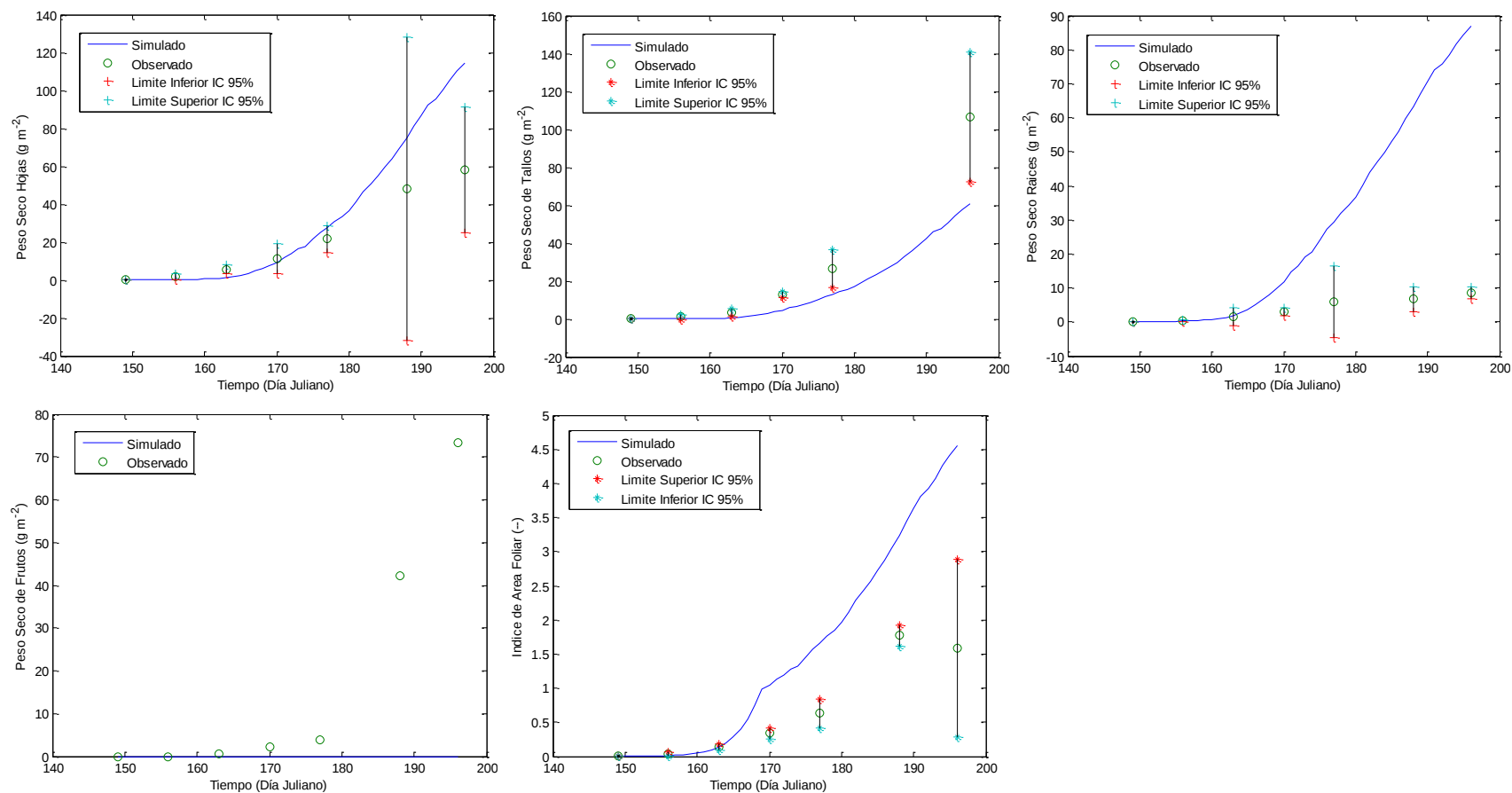


Figura 1.B. Simulaciones de las variables de estado correspondientes a la calibración de la variable peso seco total.

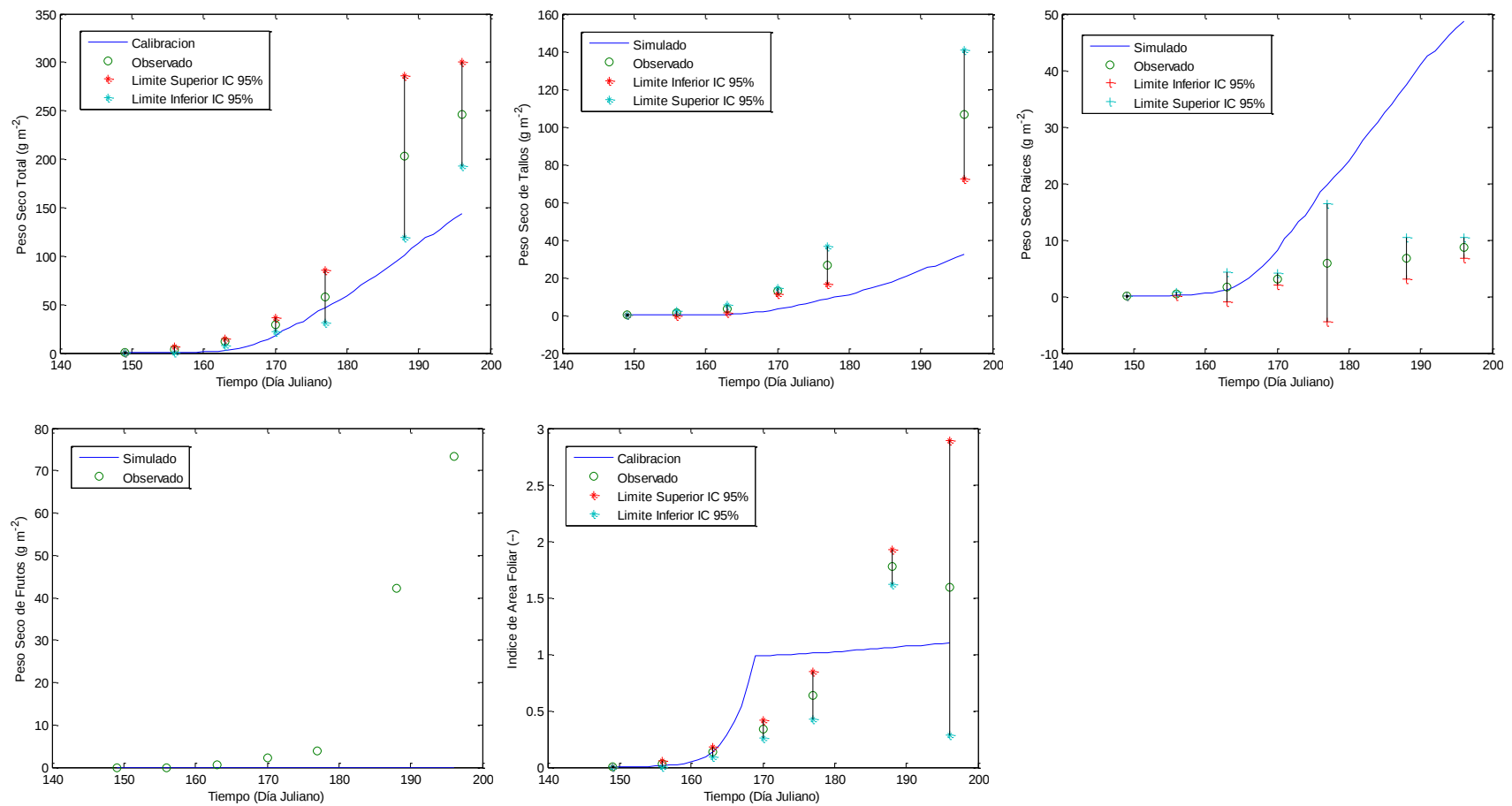


Figura 2.B. Simulaciones de las variables de estado correspondientes a la calibración de la variable peso seco de hojas.

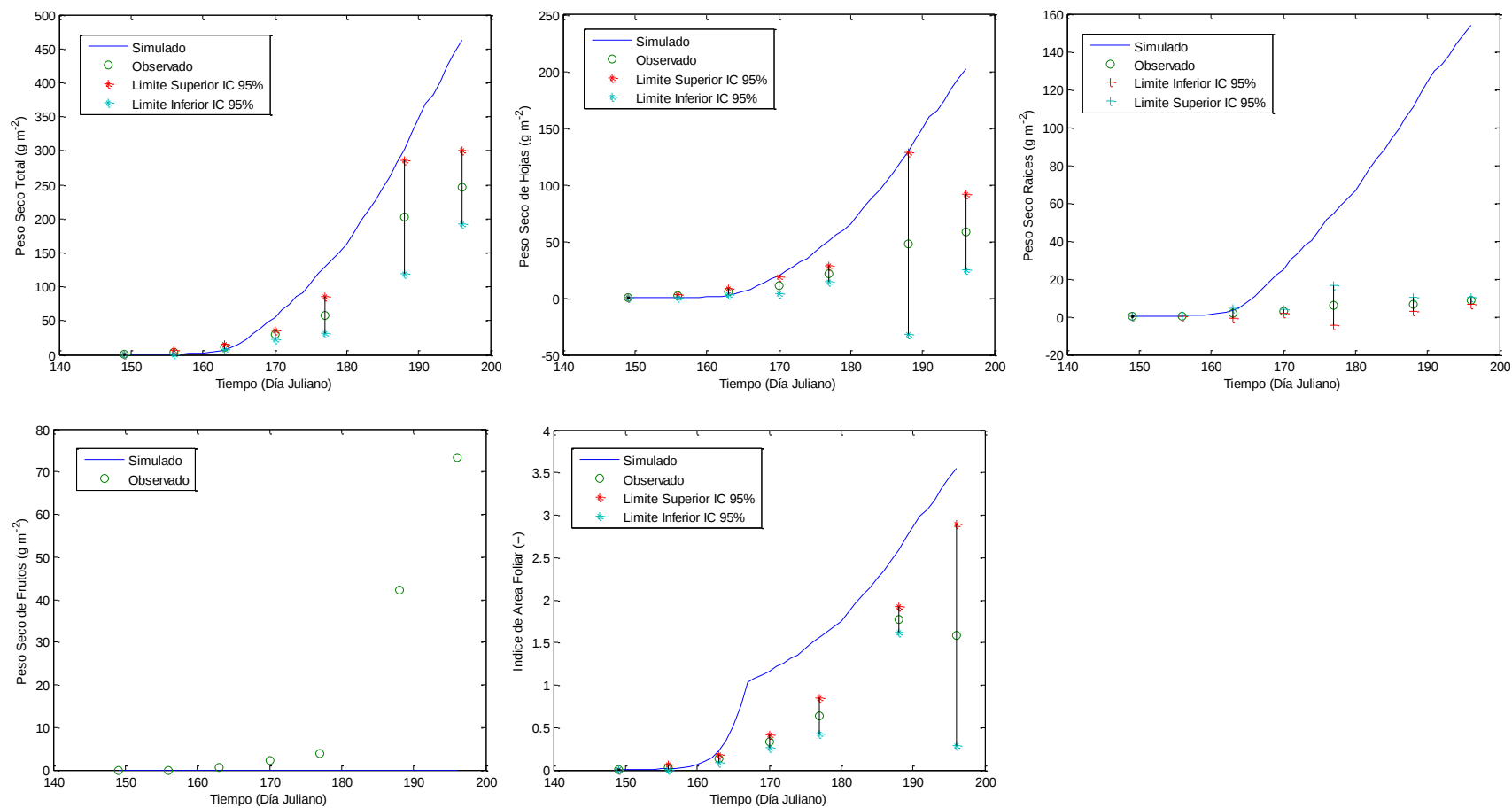


Figura 3.B. Simulaciones de las variables de estado correspondientes a la calibración de la variable peso seco de tallos.

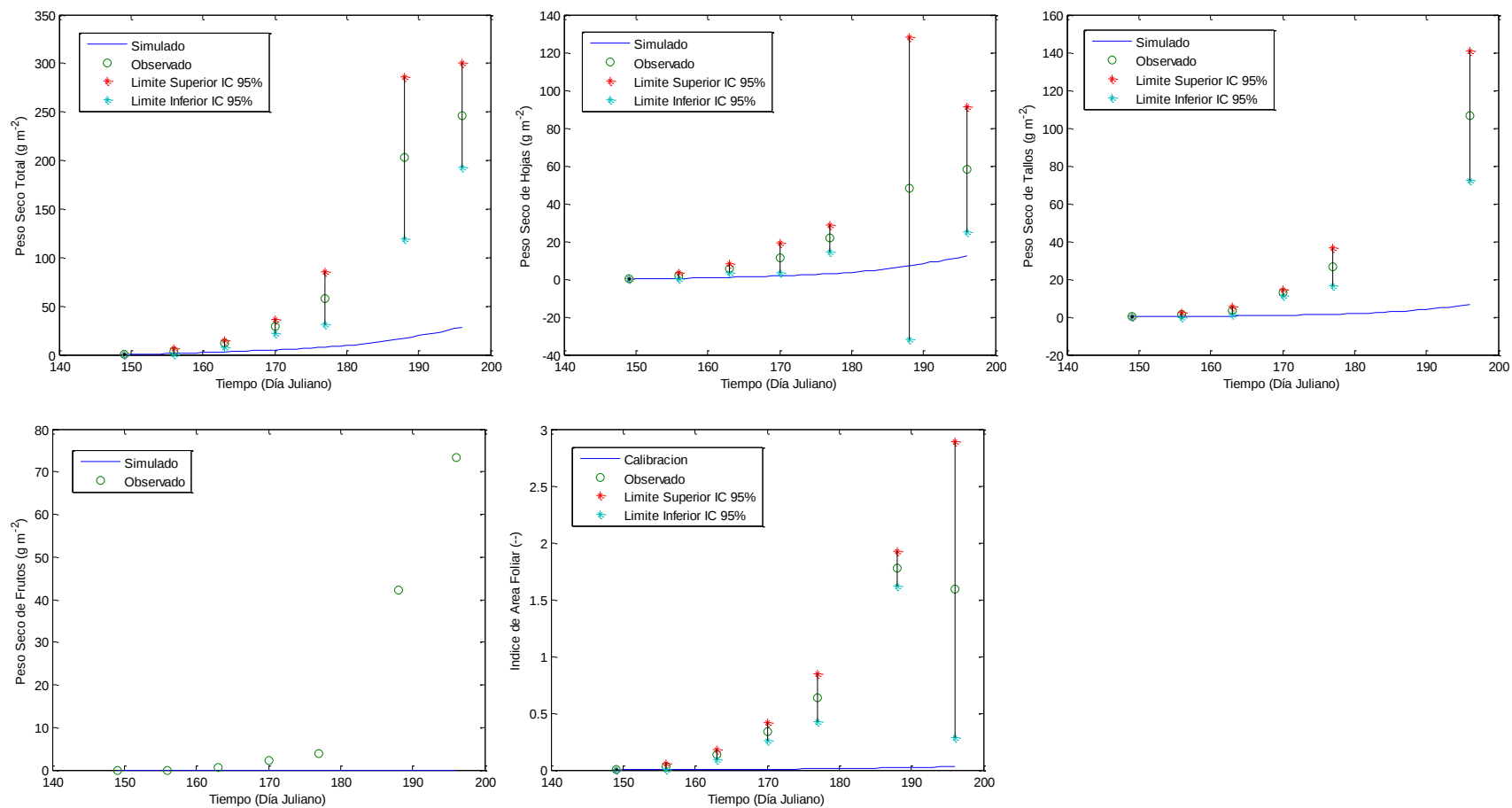


Figura 4.B. Simulaciones de las variables de estado correspondientes a la calibración de la variable peso seco de raíces.

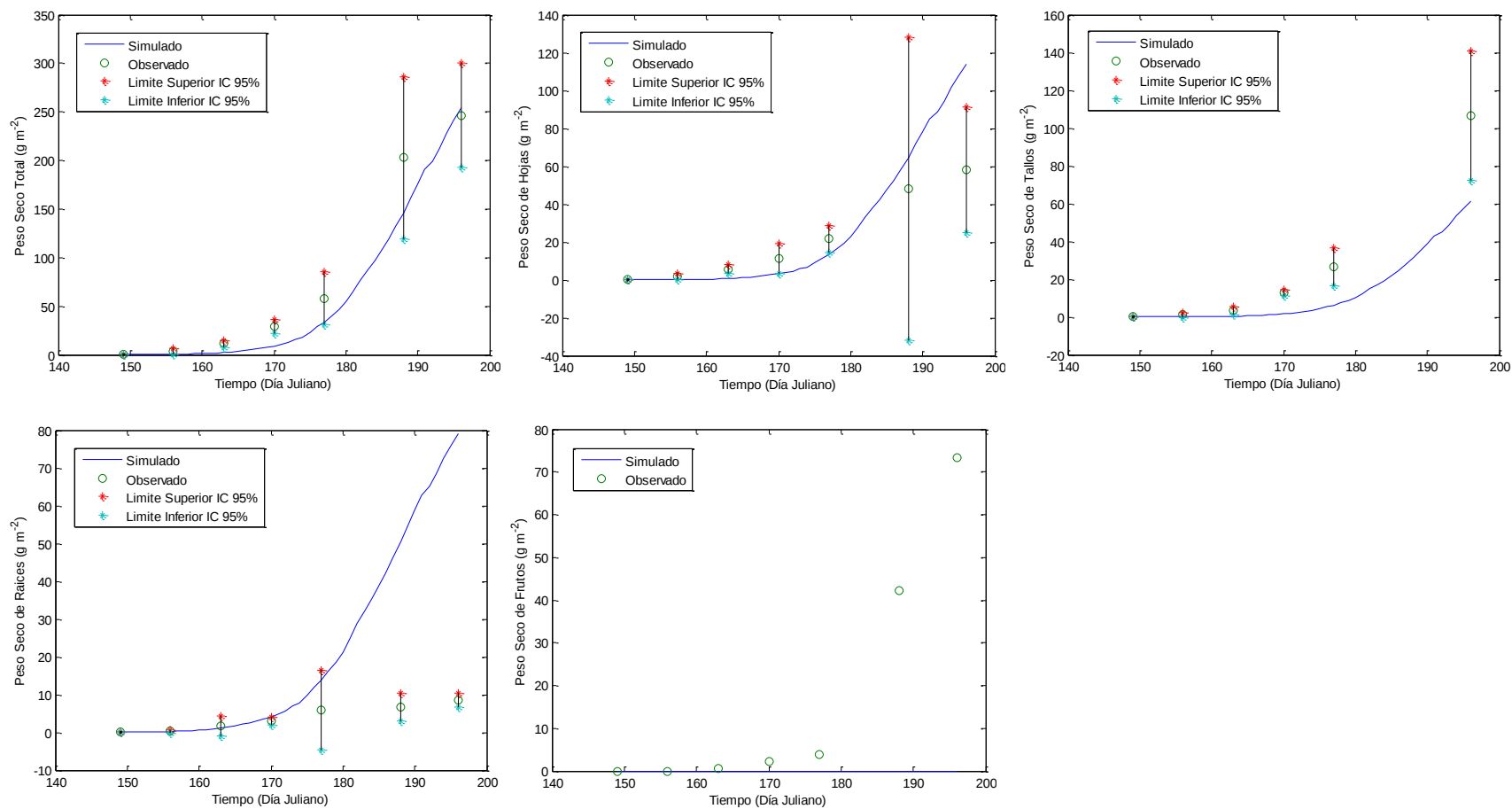


Figura 5.B. Simulaciones de las variables de estado correspondientes a la calibración de la variable índice de área foliar.

Apéndice C

NOMENCLATURA

Cuadro 1.C. Parámetros del modelo SUCROS

Variable	Símbolo	Unidades	Valor
Latitud	LATT	Grados	19.0
Densidad de plantas	NPL	Plantas m ⁻²	1.48
Día de emergencia del cultivo	DAYEM	Día	139
Valor extrapolado de área foliar a emergencia	LA0	m ² planta ⁻¹	0.00124
Tasa de crecimiento relativa durante la fase de crecimiento exponencial del área foliar	RGRL	°C ⁻¹ d ⁻¹	0.009
Temperatura base para el crecimiento de área foliar	TBASE	°C	10.0
Área foliar específica	SLA	m ² g ⁻¹	0.022
Tasa de asimilación potencial de CO ₂ a saturación de luz para hojas individuales	AMX	g m ⁻² s ⁻¹	1.11x10 ⁻³
Eficiencia inicial de uso de luz para hojas individuales	EFF	g J ⁻¹	12.5x10 ⁻⁶
Coeficiente de extinción de la radiación del follaje	KDF	m ² ha ⁻¹	0.6
Coeficiente de dispersión de las hojas para PAR.	SCP	--	0.20
Constante de la función de respiración de mantenimiento	Q10	--	2.0
Temperatura de referencia para la respiración de mantenimiento	TREF	°C	25
Coeficiente de respiración de mantenimiento para las hojas	MAINLV	g g ⁻¹ d ⁻¹	0.03
Coeficiente de respiración de mantenimiento para los tallos	MAINST	g g ⁻¹ d ⁻¹	0.015
Coeficiente de respiración de mantenimiento para las raíces	MAINRT	g g ⁻¹ d ⁻¹	0.015
Coeficiente de respiración de mantenimiento para granos	MAINSO	g g ⁻¹ d ⁻¹	0.01
Requerimiento de asimilados para la producción de biomasa foliar	ASRQLV	g g ⁻¹	1.463
Requerimiento de asimilados para la producción de biomasa de tallos	ASRQST	g g ⁻¹	1.513
Requerimiento de asimilados para la producción de biomasa de raíces	ASRQRT	g g ⁻¹	1.444
Requerimiento de asimilados para la producción de biomasa de órganos	ASRQSO	g g ⁻¹	1.415
Intervalo de tiempo de integración	DeltaT	d	1.0
Índice de área de hoja crítica más allá del cual ocurre la muerte por autosombreo	LAICR	² m ⁻²	4.0
Fracción del peso del tallo eventualmente traslocado a órganos de almacenamiento	FRTRL	--	0.2
Fracción de masa de carbono en los tallos	CFST	g g ⁻¹	0.494
Fracción de masa de carbono en las hojas	CFLV	g g ⁻¹	0.459
Fracción de masa de carbono en las raíces	CFRT	g g ⁻¹	0.467
Fracción de masa de carbono en los frutos	CFSO	g g ⁻¹	0.471
Parámetro para determinar la tasa de incremento en la tasa de muerte relativa (RDR)	FRDR	--	1.0
Factor de removilización para la removilización de los carbohidratos de los tallos en glucosa (almidón a glucosa)	CONVL	--	0.947

Cuadro 1.C. Parámetros del modelo SUCROS

Continuación...

Variable	Símbolo	Unidades	Valor
Calor latente de evaporación para el agua	LHVAP	J Kg ⁻¹	2.4x10 ⁶
Constante psicrométrica del instrumento	PSYCH	KPa °C ⁻¹	0.067
Constante de Stefan-Boltzmann	BOLTZM	J m ⁻² d ⁻¹ °K ⁻⁴	5.668x10 ⁻⁸
Contenido volumétrico de agua a saturación para la capa 1 del suelo	WCST1	cm ³ cm ⁻³	0.45
Contenido volumétrico de agua a saturación para la capa 2 del suelo	WCST2	cm ³ cm ⁻³	0.45
Contenido volumétrico de agua a saturación para la capa 3 del suelo	WCST3	cm ³ cm ⁻³	0.45
Contenido volumétrico de agua a saturación para la capa 4 del suelo	WCST4	cm ³ cm ⁻³	0.45
Contenido volumétrico de agua a capacidad de campo para la capa 1 del suelo	WCFC1	cm ³ cm ⁻³	0.296
Contenido volumétrico de agua a capacidad de campo para la capa 2 del suelo	WCFC2	cm ³ cm ⁻³	0.296
Contenido volumétrico de agua a capacidad de campo para la capa 3 del suelo	WCFC3	cm ³ cm ⁻³	0.296
Contenido volumétrico de agua a capacidad de campo para la capa 4 del suelo	WCFC4	cm ³ cm ⁻³	0.296
Parámetro en la formula de Angstrom	A1	--	0.25
Parámetro en la formula de Angstrom	B1	--	0.45
Profundidad de la capa 1 del suelo	TKL1	mm	200
Profundidad de la capa 2 del suelo	TKL2	mm	400
Profundidad de la capa 3 del suelo	TKL3	mm	600
Profundidad de la capa 4 del suelo	TKL4	mm	800
Contenido volumétrico de agua a punto de marchitez para la capa 1 del suelo	WCWP1	cm ³ cm ⁻³	0.165
Contenido volumétrico de agua a punto de marchitez para la capa 2 del suelo	WCWP2	cm ³ cm ⁻³	0.165
Contenido volumétrico de agua a punto de marchitez para la capa 3 del suelo	WCWP3	cm ³ cm ⁻³	0.165
Contenido volumétrico de agua a punto de marchitez para la capa 4 del suelo	WCWP4	cm ³ cm ⁻³	0.165
Contenido volumétrico de agua para la capa 1 donde el agua comienza a registrarse	WCWET1	cm ³ cm ⁻³	0.35
Contenido volumétrico de agua para la capa 2 donde el agua comienza a registrarse	WCWET2	cm ³ cm ⁻³	0.35
Contenido volumétrico de agua para la capa 3 donde el agua comienza a registrarse	WCWET3	cm ³ cm ⁻³	0.35
Contenido volumétrico de agua para la capa 4 donde el agua comienza a registrarse	WCWET4	cm ³ cm ⁻³	0.35
Capacidad de intercepción de la precipitación de una capa de hojas	INTC	mm d ⁻¹	0.25
Tasa máxima de drenaje del subsuelo	MDRATE	mm d ⁻¹	50.0
Contenido volumétrico de agua a aire seco para la capa 1 del suelo	WCAD1	cm ³ cm ⁻³	0.025
Contenido volumétrico de agua a aire seco para la capa 2 del suelo	WCAD2	cm ³ cm ⁻³	0.025
Contenido volumétrico de agua a aire seco para la capa 3 del suelo	WCAD3	cm ³ cm ⁻³	0.025
Contenido volumétrico de agua a aire seco para la capa 4 del suelo	WCAD4	cm ³ cm ⁻³	0.025
Coeficiente específico de extinción del suelo	EES	mm ⁻¹	0.002
Tasa de transpiración potencial característica	TRANSC	mm d ⁻¹	9.0

Cuadro 2.C. Parámetros del modelo LINTUL1

Variable	Símbolo	Unidades	Valor
Índice de área foliar inicial (a la emergencia del cultivo)	LAI	$\text{m}^2 \text{ m}^{-2}$	0.012
Área foliar específica	SLA	$\text{m}^2 \text{ g}^{-1}$	0.022
Temperatura base	TBASE	$^{\circ}\text{C}$	10.0
Día de emergencia del cultivo	DAYEM	d	139
Tasa relativa de crecimiento de LAI durante la fase exponencial	RGRL	$^{\circ}\text{C}^{-1} \text{ d}^{-1}$	0.009
Suma de temperatura hasta la anthesis	TSUMAN	$^{\circ}\text{C d}$	1110
Índice de área de hoja crítica más allá del cual ocurre la muerte por autosombreo	LAICR	$\text{m}^2 \text{ m}^{-2}$	4.0
Tasa máxima de muerte relativa de las hojas debido al autosombreo	RDRSHM	d^{-1}	0.03
Eficiencia del uso de la luz (materia seca producida por unidad de radiación fotosintéticamente activa interceptada (PAR))	LUE	g MJ^{-1}	3.0
Coefficiente de extinción para la radiación fotosintéticamente activa	K	--	0.6
Intervalo de paso de integración	DeltaT	d	1.0

Apéndice D

Implementación C-Mex s-function del modelo SUCROS971 para la simulación del crecimiento potencial del cultivo de tomate de cáscara

```
1  /*
2  Irineo L. Lopez Cruz
3  Postgrado IAUIA, UACH
4  Chapingo Mex.
5  2006-2008
6  */

7  #define S_FUNCTION_LEVEL 2
8  #define S_FUNCTION_NAME sucros971

9  /*
10 * Need to include simstruc.h for the definition of the SimStruct and
11 * its associated macro definitions.
12 */

13 #include "simstruc.h"
14 #include <math.h>
15 #include <stdlib.h>
16 #define ROWS 50
17 #define COLS 2

18 #define max(A,B) ((A) > (B) ? (A) : (B))
19 #define min(A,B) ((A) < (B) ? (A) : (B))
20 #define u(element)(*uPtrs[element])

21 /* Funciones default:FST*/
22 double minterp1(double,double,double,double,double);
23 double mafgen(double z[][COLS],double,int);
24 double limit(double, double,double );
25 double NotNull(double );
26 double max1(double ,double , double , double );
27 double insw(double ,double ,double );

28 /* Funciones del modelo SUCROS */
29 void gla(double Time,double Doyem,double DTeff,double Dvs,double Rgrl,double Deltat,double Sla,double
30 Lai,double Glv,double *CrecLai);
31 void Astro(double Doy, double Lat, double *Sc,double *Ds0, double *Sinld, double *Cosld, double *Dayl,
32 double *Dsinb, double *Dsinbe);
33 void Assim(double Scp, double Amax,double Eff,double Kdf, double Lai, double Sinb, double Pardr, double
34 Pardf,double *Fgros);
35 void Totass(double Day, double Lat, double Dtr, double Scp, double Amax, double Eff, double Kdf, double
36 Lai, double *Dtga);

37 /*Used to read the parameter number 1*/
38 #define x10 p[0]
39 #define x20 p[1]
40 #define x30 p[2]
41 #define x40 p[3]
42 #define x50 p[4]
43 #define x60 p[5]
44 #define x70 p[6]
45 #define x80 p[7]
46 #define LATT p[8]
47 #define NPL p[9]
48 #define DAYEM p[10]
49 #define LA0 p[11]
```

```

50 #define RGRL          p[12]
51 #define TBASE         p[13]
52 #define SLA           p[14]
53 #define AMX           p[15]
54 #define EFF           p[16]
55 #define KDF           p[17]
56 #define SCP           p[18]
57 #define Q10           p[19]
58 #define TREF          p[20]
59 #define MAINLV        p[21]
60 #define MAINST        p[22]
61 #define MAINRT        p[23]
62 #define MAINSO        p[24]
63 #define ASRQLV        p[25]
64 #define ASRQST        p[26]
65 #define ASRQRT        p[27]
66 #define ASRQSO        p[28]
67 #define DeltaT        p[29]
68 #define LAICR         p[30]
69 #define FRTRL         p[31]
70 #define CFST          p[32]
71 #define CFLV          p[33]
72 #define CFRT          p[34]
73 #define CFSO          p[35]
74 #define FRDR          p[36]
75 #define CONVL         p[37]

76 /* Error handling
77  * -----
78  *
79  * You should use the following technique to report errors encountered within
80  * an S-function:
81  *
82  *     ssSetErrorStatus(S,"Error encountered due to ...");
83  *     return;
84  *
85  * Note that the 2nd argument to ssSetErrorStatus must be persistent memory.
86  * It cannot be a local variable. For example the following will cause
87  * unpredictable errors:
88  *
89  *     mdlOutputs()
90  *     {
91  *         char msg[256];      {ILLEGAL: to fix use "static char msg[256];"}
92  *         sprintf(msg,"Error due to %s", string);
93  *         ssSetErrorStatus(S,msg);
94  *         return;
95  *     }
96  *
97  * See matlabroot/simulink/src/sfunctmpl.doc for more details.
98  */

99 /*=====
100  * S-function methods *
101  *=====*/

102 /* Function: mdlInitializeSizes =====
103  * Abstract:
104  * The sizes information is used by Simulink to determine the S-function
105  * block's characteristics (number of inputs, outputs, states, etc.).
106  */
107 static void mdlInitializeSizes(SimStruct *S)
108 {

```

```

109      /* See sfuntmpl.doc for more details on the macros below */

110      ssSetNumSFcnParams(S, 1); /* Number of expected parameters */
111      if (ssGetNumSFcnParams(S) != ssGetSFcnParamsCount(S)) {
112          /* Return if number of expected != number of actual parameters */
113          return;
114      }

115      ssSetNumContStates(S, 8);
116      ssSetNumDiscStates(S, 0);

117      if (!ssSetNumInputPorts(S, 1)) return;
118      /*Entradas del modelo*/
119      ssSetInputPortWidth(S, 0, 3);

120      ssSetInputPortDirectFeedThrough(S, 0, 0);

121      if (!ssSetNumOutputPorts(S, 1)) return;
122      ssSetOutputPortWidth(S, 0, 10);

123      ssSetNumSampleTimes(S, 1);
124      ssSetNumRWork(S, 0);
125      ssSetNumIWork(S, 0);
126      ssSetNumPWork(S, 0);
127      ssSetNumModes(S, 0);
128      ssSetNumNonsampledZCs(S, 0);

129      ssSetOptions(S, 0);
130  }

131  /* Function: mdlInitializeSampleTimes =====
132  * Abstract:
133  *   This function is used to specify the sample time(s) for your
134  *   S-function. You must register the same number of sample times as
135  *   specified in ssSetNumSampleTimes.
136  */
137  static void mdlInitializeSampleTimes(SimStruct *S)
138  {
139      ssSetSampleTime(S, 0, CONTINUOUS_SAMPLE_TIME);
140      ssSetOffsetTime(S, 0, 0.0);
141  }

142  #define MDL_INITIALIZE_CONDITIONS /* Change to #undef to remove function */
143  #if defined(MDL_INITIALIZE_CONDITIONS)
144      /* Function: mdlInitializeConditions =====
145      * Abstract:
146      *   In this function, you should initialize the continuous and discrete
147      *   states for your S-function block. The initial states are placed
148      *   in the state vector, ssGetContStates(S) or ssGetRealDiscStates(S).
149      *   You can also perform any other initialization activities that your
150      *   S-function may require. Note, this routine will be called at the
151      *   start of simulation and if it is present in an enabled subsystem
152      *   configured to reset states, it will be call when the enabled subsystem
153      *   restarts execution to reset the states.
154      */
155      static void mdlInitializeConditions(SimStruct *S)
156      {
157          real_T *x0=ssGetContStates(S);
158          double *p=mxGetPr(ssGetSFcnParam(S,0));
159          x0[0]=x10;
160          x0[1]=x20;
161          x0[2]=x30;

```

```

162         x0[3]=x40;
163         x0[4]=x50;
164         x0[5]=x60;
165         x0[6]=x70;
166         x0[7]=x80;
167     }
168 #endif /* MDL_INITIALIZE_CONDITIONS */

169 #define MDL_START /* Change to #undef to remove function */
170 #if defined(MDL_START)
171     /* Function: mdlStart =====
172     * Abstract:
173     * This function is called once at start of model execution. If you
174     * have states that should be initialized once, this is the place
175     * to do it.
176     */
177     static void mdlStart(SimStruct *S)
178     {
179     }
180 #endif /* MDL_START */

181 /* Function: mdlOutputs =====
182 * Abstract:
183 * In this function, you compute the outputs of your S-function
184 * block. Generally outputs are placed in the output vector, ssGetY(S).
185 */
186 static void mdlOutputs(SimStruct *S, int_T tid)
187 {

188     double DvrVt[3][2];
189     double DvrRt[3][2];
190     double FlvTb[7][2];
191     double FshTb[14][2];
192     double Fsttb[8][2];
193     double Amdvst[4][2];
194     double Amtmpt[6][2];

195     real_T *y=ssGetOutputPortRealSignal(S,0);
196     real_T *x=ssGetContStates(S);
197     InputRealPtrsType uPtrs=ssGetInputPortRealSignalPtrs(S,0);
198     double *p=mxGetPr(ssGetSFcnParam(S,0));
199     real_T t=ssGetT(S);

200     double DevRate,Emergencia,Dia, DtMax,DtMin,Dtr,DavTemp,RLeafAreaIndex;
201     double CrecLAI,MuerteLAI,GlV,Gsh,Gst, Gso,Grt, Rdr,RdrDv,RdrSh,RdrLt,RdrSl;
202     double Fsh,Flv,Fst,Fso,Frt,Gtw, GPhot,Maint,Asrq,Dtga,Dso, Maints,Teff,Mndvs;
203     double Wlv,RDlv, RWlvG, Amax,Amdvs,Amtmp,DdTmp,Dteff;
204     double Dayl, Transl;
205     double Tnassi,Chkin,Chkfl, Rtnass,CO2Rt,CO2Lv,CO2St, CO2So,Chkdif;

206     /*
207     x[0] : Dvs
208     x[1] : Lai
209     x[2] : wlvG
210     x[3] : wlvD
211     x[4] : wrt
212     x[5] : wst
213     x[6] : wso
214     x[7] : Tnass
215     */

216     y[0]=x[0];

```

```

217 y[1]=x[1];
218 y[2]=x[2];
219 y[3]=x[3];
220 y[4]=x[4];
221 y[5]=x[5];
222 y[6]=x[6];
223 /* biomasa foliar*/
224 y[7]=x[2]+x[3];
225 /*biomasa de la parte aerea */
226 y[8]=x[2]+x[3]+x[5]+x[6];
227 /*biomasa total incluyendo raices*/
228 y[9]=x[2]+x[3]+x[4]+x[5]+x[6];
229 }

230 #define MDL_UPDATE /* Change to #undef to remove function */
231 #if defined(MDL_UPDATE)
232 /* Function: mdlUpdate =====
233  * Abstract:
234  * This function is called once for every major integration time step.
235  * Discrete states are typically updated here, but this function is useful
236  * for performing any tasks that should only take place once per
237  * integration step.
238  */
239 static void mdlUpdate(SimStruct *S, int_T tid)
240 {
241 }
242 #endif /* MDL_UPDATE */

243 #define MDL_DERIVATIVES /* Change to #undef to remove function */
244 #if defined(MDL_DERIVATIVES)
245 /* Function: mdlDerivatives =====
246  * Abstract:
247  * In this function, you compute the S-function block's derivatives.
248  * The derivatives are placed in the derivative vector, ssGetdX(S).
249  */
250 static void mdlDerivatives(SimStruct *S)
251 {
252 double DvrVt[3][2];
253 double DvrRt[3][2];
254 double FlvTb[7][2];
255 double FshTb[14][2];
256 double Fsttb[8][2];
257 double Amdvst[4][2];
258 double Amtmpt[6][2];
259 double FsoTb[4][2];

260 real_T *dx=ssGetdX(S);
261 real_T *x=ssGetContStates(S);
262 InputRealPtrsType uPtrs=ssGetInputPortRealSignalPtrs(S,0);
263 double *p=mxGetPr(ssGetSFcnParam(S,0));
264 real_T t=ssGetT(S);

265 double DevRate,Emergencia,Dia, DtMax,DtMin,Dtr,DavTemp,RLeafAreaIndex;
266 double CrecLAI,MuerteLAI,GlV,Gsh,Gst, Gso,Grt, Rdr,RdrDv,RdrSh,RdrLt,RdrSl;
267 double Fsh,Flv,Fst,Fso,Frt,Gtw, GPhot,Maint,Asrq,Dtga, Maints,Teff,Mndvs;
268 double Wlv,RDlv, RWlvG, Amax,Amdvs,Amtmp,DdTmp,Dteff;
269 double Transl,Dlv;
270 double Tnassi,Chkin,Chkfl, Rtnass,CO2Rt,CO2Lv,CO2St, CO2So,Chkdif;

271 /*
272 Variables de entrada del modelo
273 u(0): Temperatura minima

```

```

274 u(1): Temperatura maxima
275 u(2): Radiacion solar

276 x[0]: Estado de desarrollo (DVS)
277 x[1]: IAF (IAF)
278 x[2]: biomasa de hojas verdes (wlv)
279 x[3]: biomasa de hojas secas(wld)
280 x[4]: biomasa de raices(wrt)
281 x[5]: biomasa de tallos(wst)
282 x[6]: biomasa de organos(wso)
283 */

284 /* Tabla de tasa de desarrollo */
285 DvrVt[0][0]=-10.0;
286 DvrVt[1][0]=0;
287 DvrVt[2][0]=30;
288 DvrVt[0][1]=0;
289 DvrVt[1][1]=0;
290 DvrVt[2][1]=0.027;

291 DvrRt[0][0]=-10.0;
292 DvrRt[1][0]=0;
293 DvrRt[2][0]=30;
294 DvrRt[0][1]=0;
295 DvrRt[1][1]=0;
296 DvrRt[2][1]=0.031;
297 /* Tablas de division de la biomasa en diferentes organos */
298 FlvTb[0][0]=0.00;
299 FlvTb[1][0]=0.10;
300 FlvTb[2][0]=0.25;
301 FlvTb[3][0]=0.50;
302 FlvTb[4][0]=0.70;
303 FlvTb[5][0]=0.95;
304 FlvTb[6][0]=2.50;

305 FlvTb[0][1]=0.65;
306 FlvTb[1][1]=0.65;
307 FlvTb[2][1]=0.70;
308 FlvTb[3][1]=0.50;
309 FlvTb[4][1]=0.15;
310 FlvTb[5][1]=0.00;
311 FlvTb[6][1]=0.00;

312 FshTb[0][0]=0.00;
313 FshTb[1][0]=0.10;
314 FshTb[2][0]=0.20;
315 FshTb[3][0]=0.35;
316 FshTb[4][0]=0.40;
317 FshTb[5][0]=0.50;
318 FshTb[6][0]=0.60;
319 FshTb[7][0]=0.70;
320 FshTb[8][0]=0.80;
321 FshTb[9][0]=0.90;
322 FshTb[10][0]=1.00;
323 FshTb[11][0]=1.10;
324 FshTb[12][0]=1.20;
325 FshTb[13][0]=2.50;

326 FshTb[0][1]=0.50;
327 FshTb[1][1]=0.50;
328 FshTb[2][1]=0.60;
329 FshTb[3][1]=0.78;

```

```

330     FshTb[4][1]=0.83;
331     FshTb[5][1]=0.87;
332     FshTb[6][1]=0.90;
333     FshTb[7][1]=0.93;
334     FshTb[8][1]=0.95;
335     FshTb[9][1]=0.97;
336     FshTb[10][1]=0.98;
337     FshTb[11][1]=0.99;
338     FshTb[12][1]=1.0;
339     FshTb[13][1]=1.0;

340     Fsttb[0][0]=0.00;
341     Fsttb[1][0]=0.10;
342     Fsttb[2][0]=0.25;
343     Fsttb[3][0]=0.50;
344     Fsttb[4][0]=0.70;
345     Fsttb[5][0]=0.95;
346     Fsttb[6][0]=1.05;
347     Fsttb[7][0]=2.50;

348     Fsttb[0][1]=0.35;
349     Fsttb[1][1]=0.35;
350     Fsttb[2][1]=0.30;
351     Fsttb[3][1]=0.50;
352     Fsttb[4][1]=0.85;
353     Fsttb[5][1]=1.00;
354     Fsttb[6][1]=0.00;
355     Fsttb[7][1]=0.00;

356     FsoTb[0][0]=0.00;
357     FsoTb[1][0]=0.95;
358     FsoTb[2][0]=1.05;
359     FsoTb[3][0]=2.50;

360     FsoTb[0][1]=0.00;
361     FsoTb[1][1]=0.00;
362     FsoTb[2][1]=1.00;
363     FsoTb[3][1]=1.00;

364     FsoTb[0][1]=0.00;
365     FsoTb[1][1]=1.00;
366     FsoTb[2][1]=1.00;
367     FsoTb[3][1]=1.00;

368     /*Tablas de asimilacion de CO2 */
369     Amdvst[0][0]=0.0;
370     Amdvst[1][0]=1.0;
371     Amdvst[2][0]=2.0;
372     Amdvst[3][0]=2.5;

373     Amdvst[0][1]=1.0;
374     Amdvst[1][1]=1.0;
375     Amdvst[2][1]=0.5;
376     Amdvst[3][1]=0.0;

377     Amtmpt[0][0]=-10.00;
378     Amtmpt[1][0]=0.00;
379     Amtmpt[2][0]=10.00;
380     Amtmpt[3][0]=25.00;
381     Amtmpt[4][0]=35.00;
382     Amtmpt[5][0]=50.00;

```

```

383 Amtmpt[0][1]=0.00;
384 Amtmpt[1][1]=0.00;
385 Amtmpt[2][1]=1.00;
386 Amtmpt[3][1]=1.00;
387 Amtmpt[4][1]=0.00;
388 Amtmpt[5][1]=0.00;

389 /* Dia = 1.0+fmod(t-1,365.0);*/
390 Dia =fmod(t,365.0);
391 Emergencia=insw(t-DAYEM,0.0,1.0);

392 DtMax=u(0);
393 DtMin=u(1);
394 Dtr=u(2)*1000000.0;

395 DavTemp=0.5*(DtMax+DtMin);
396 DevRate=insw(x[0]-1.0,mafgen(DvrVt,DavTemp,3),mafgen(DvrRt,DavTemp,3))*Emergencia;
397 RdrSh=limit(0.0,0.03,0.03*(x[1]-LAICR)/LAICR);
398 RdrDv=insw(x[0]-1.0,0.0,DevRate/(max(0.1,2.0-x[0]))*FRDR);
399 Rdr=max(RdrDv,RdrSh);
400 MuerteLAI=x[1]*Rdr;
401 /* Tasa de fotosintesis a saturacion de luz */

402 DdTmp=DtMax-0.25*(DtMax-DtMin);
403 Amdvs=mafgen(Amdvst,x[0],4);
404 Amtmp=mafgen(Amtmpt,DdTmp,6);
405 Amax=AMX*Amdvs*Amtmp*Emergencia;
406 Totass(Dia,LATT,Dtr,SCP,Amax,EFF,KDF,x[1],&Dtga);

407 GPhot=Dtga*30.0/44.0;
408 Wlv=x[2]+x[3];
409 Mndvs=x[2]/(NotNull(Wlv));
410 Teff=pow(Q10,(DavTemp-TREF)/10.0);

411 Maints=MAINLV*x[2]+MAINST*x[5]+MAINRT*x[4]+MAINSO*x[6];
412 Maint=Maints*Teff*Mndvs*Emergencia;

413 /*Particionamiento de la biomasa */
414 Fsh=mafgen(FshTb,x[0],14);
415 Flv=mafgen(FlvTb,x[0],7);
416 Fst=mafgen(Fsttb,x[0],8);
417 Fso=mafgen(FsoTb,x[0],4);

418 Frt=1.0-Fsh;
419 Asrq=Fsh*(ASRQLV*Flv+ASRQST*Fst+ASRQSO*Fso)+ASRQRT*Frt;
420 Transl = insw(x[0]-1.0,0.0,x[5]*DevRate*FRTRL);
421 Gtw=(GPhot-Maint+CONVL*Transl*CFST*30.0/12.0)/Asrq;
422 /* Tasa de crecimiento total, tallos y raices*/

423 Glv=Flv*Fsh*Gtw;
424 Gst=Fst*Fsh*Gtw-Transl;
425 Gso=Fso*Fsh*Gtw;
426 Grt=Frt*Gtw;
427 Dteff=max(0.0,DavTemp-TBASE);
428 gla(t,DAYEM,Dteff,x[0],RGRL,DeltaT,SLA,x[1],Glv,&CrecLAI);
429 RLeafAreaIndex=CrecLAI-MuerteLAI;
430 Dlv=x[2]*MuerteLAI/(NotNull(x[1]));
431 RWlvG=Glv-Dlv;

432 Chkin=Wlv*CFLV+x[5]*CFST+x[4]*CFRT+x[6]*CFSO;
433 Chkfl=x[7]*(12.0/44.0);
434 CO2Rt=44.0/12.0*(ASRQRT*12.0/30.0-CFRT);

```

```

435 CO2Lv=44.0/12.0*(ASRQLV*12.0/30.0-CFLV);
436 CO2St=44.0/12.0*(ASRQST*12.0/30.0-CFST);
437 CO2So=44.0/12.0*(ASRQSO*12.0/30.0-CFSO);
438 Chkdif=(Chkin-Chkfl)/NotNull(Chkin);
439 Rtnass=((GPhot-Maint)*44.0/30.0)-(Grt*CO2Rt+Glv*CO2Lv+(Gst+Transl)*CO2St+Gso*CO2So+(1.0-
440 CONVL)*Transl*CFST*44.0/12.0);

441 /*
442 x[0] : Dvs
443 x[1] : Lai
444 x[2] : wlvlg
445 x[3] : wlvld
446 x[4] : wrt
447 x[5] : wst
448 x[6] : wso
449 x[7] : Tnass
450 */

451 dx[0]=DevRate;
452 dx[1]=RLeafAreaIndex;
453 dx[2]=RWlvlg;
454 dx[3]=Dlv;
455 dx[4]=Grt;
456 dx[5]=Gst;
457 dx[6]=Gso;
458 dx[7]=Rtnass;

459 }
460 #endif /* MDL_DERIVATIVES */

461 /* Function: mdlTerminate =====
462 * Abstract:
463 * In this function, you should perform any actions that are necessary
464 * at the termination of a simulation. For example, if memory was
465 * allocated in mdlInitializeConditions, this is the place to free it.
466 */
467 static void mdlTerminate(SimStruct *S)
468 {
469 }

470 double mafgen(double z[][COLS],double x, int nrows)
471 {
472 int i;
473 double y;

474 for (i=0;i<nrows;i++){
475 if (z[i][0]==x){
476 y=z[i][1];
477 break;
478 }
479 else if ( ( z[i][0]< x) && (z[i+1][0]> x)){
480 y=minterp1(z[i][0],z[i][1],z[i+1][0],z[i+1][1],x);
481 break;
482 }
483 }
484 return(y);
485 }

486 double minterp1(double x1,double y1,double x2,double y2,double x)
487 {
488 double y;
489 y=y1+(y2-y1)*(x-x1)/(x2-x1);

```

```

490     return (y);
491 }

492 double insw(double x,double y1,double y2)
493 {
494     double y;
495     if (x<0)
496         y=y1;
497     else
498         y=y2;
499     return (y);
500 }

501 /* Subrutinas SUCROS97 Goudriaan 1994*/
502 void gla(double Time,double Doyem,double DTeff,double Dvs,double Rgrl,double Deltat,double Sla,double
503 Lai,double Glv,double *CrecLai)
504 {

505     *CrecLai =Sla*Glv;
506     if ((Dvs < 0.3) && (Lai < 0.75))
507         *CrecLai=(Lai*(exp(Rgrl*DTeff*Deltat)-1.0))/Deltat;

508     if (Time <= Doyem)
509         *CrecLai=0.0;
510 }

511 void Assim(double Scp, double Amax,double Eff,double Kdf, double Lai, double Sinb, double Pardr, double
512 Pardf,double *Fgros)
513 {

514     double xGauss[5],wGauss[5];
515     double Sqv, Refh, Refs, Clustf, Kbl, Kdrt, Laic, Visdf, Vist, Visd, Visshd, Fgrsh, Vispp, Fgrsun, Vissun, Fgrs,
516     Fsla, Fgl;
517     int i,j;
518     xGauss[0]=0.0469101;
519     xGauss[1]=0.2307534;
520     xGauss[2]=0.5;
521     xGauss[3]=0.7692465;
522     xGauss[4]=0.9530899;
523     wGauss[0]=0.1184635;
524     wGauss[1]=0.2393144;
525     wGauss[2]=0.2844444;
526     wGauss[3]=0.2393144;
527     wGauss[4]=0.1184635;

528     Sqv=sqrt(1-Scp);
529     Refh=(1.0-Sqv)/(1.0+Sqv);
530     Refs=Refh*2.0/(1.0+2.0*Sinb);

531     Clustf=Kdf/(0.8*Sqv);
532     Kbl=(0.5/Sinb)*Clustf;
533     Kdrt=Kbl*Sqv;

534     *Fgros=0.0;
535     for (i=0;i<5;i++){
536         Laic=Lai*xGauss[i];
537         Visdf=(1.0-Refh)*Pardf*Kdf*exp(-Kdf*Laic);
538         Vist=(1.0-Refs)*Pardr*Kdrt*exp(-Kdrt*Laic);
539         Visd=(1.0-Scp)*Pardr*Kbl*exp(-Kbl*Laic);
540         Visshd=Visdf+Vist-Visd;
541         if (Amax > 0.0)
542             Fgrsh=Amax*(1.0-exp(-Visshd*Eff/Amax));

```

```

543     else
544         Fgrsh=0.0;
545         Vispp=(1.0-Scp)*Pardr/Sinb;
546         Fgrsun=0.0;
547         for (j=0;j<5;j++){
548             Vissun=Visshd+Vispp*xGauss[j];
549             if (Amax > 0.0)
550                 Fgrs=Amax*(1.0-exp(-Vissun*Eff/Amax));
551             else
552                 Fgrs=0.0;
553
554             Fgrsun=Fgrsun+Fgrs*wGauss[j];
555         }
556         Fslla=Clustf*exp(-Kbl*Laic);
557         Fgl=Fslla*Fgrsun+(1.0-Fslla)*Fgrsh;
558
559         *Fgros=(*Fgros)+Fgl*wGauss[i];
560     }
561     *Fgros=(*Fgros)*Lai;
562 }
563
564 void Astro(double Day, double Lat, double *Sc,double *Ds0, double *Sinld, double *Cosld, double *Dayl,
565 double *Dsinb, double *Dsinbe)
566 {
567     double PI=3.141592654;
568     double Radianes=PI/180.0;
569     double Aob, Aux2,Aux3;
570     double Dec;
571
572     Dec=-asin(sin(23.45*Radianes)* cos(2.0*PI*(Day+10.0)/365.0));
573     *Sinld= sin(Lat*Radianes)*sin(Dec);
574     *Cosld=cos(Lat*Radianes)*cos(Dec);
575     Aob=(*Sinld)/(*Cosld);
576     Aux2=(*Sinld)*(*Sinld);
577     Aux3=(*Cosld)*(*Cosld);
578     *Dayl = 12.0 * (1.0+2.0 * asin(Aob)/PI);
579     *Dsinb=3600.0*((*Dayl)*(*Sinld)+24.0*(*Cosld)*sqrt(1.0-Aob*Aob)/PI);
580     *Dsinbe=3600.0*((*Dayl)*((*Sinld)+0.4*(Aux2+0.5*Aux3)) + 12.0*(*Cosld)*(2.0+3.0*0.4*(*Sinld))*sqrt(1.0-
581     Aob*Aob)/PI);
582     *Sc = 1370.0 * (1.+0.033*cos(2.0*PI*Day/365.0));
583     *Ds0=(*Sc) * (*Dsinb);
584
585 }
586
587 void Totass(double Day, double Lat, double Dtr, double Scp, double Amax, double Eff, double Kdf, double
588 Lai, double *Dtga)
589 {
590     double xGauss[3], wGauss[3];
591     double Sc,Sinld,Cosld,Dsinb,Dsinbe,Sinb,Pard,Pardf;
592     double Par, PI, Dayl, Ds0;
593     double Hora, Atmtr,Frdf,Pardr, Fgros, Dtgas;
594     int i;
595     PI =3.141592654;
596
597     xGauss[0]=0.112702;
598     xGauss[1]=0.50;
599     xGauss[2]=0.887298;
600     wGauss[0]=0.277778;
601     wGauss[1]=0.444444;

```

```

595     wGauss[2]=0.277778;

596     Astro(Day,Lat,&Sc,&Ds0,&SinId,&CosId,&Dayl,&Dsinb,&Dsinbe);
597     Dtgas=0.0;
598     for (i=0;i<3;i++){

599         Hora=12.0+0.5*Dayl*xGauss[i];
600         Sinb=max(0.0,SinId+CosId*cos(2.0*PI*(Hora+12.0)/24.0));
601         Par=0.5*Dtr*Sinb*(1.0+0.4*Sinb)/Dsinbe;
602         Atmtr=Par/(0.5*Sc*Sinb);
603         if (Atmtr <= 0.22)
604             Frdf=1.0;
605         else if (Atmtr > 0.22 && (Atmtr <= 0.35) )
606             Frdf=1.0-6.4*pow((Atmtr-0.22),2.0);
607         else
608             Frdf=1.47-1.66*Atmtr;
609         Frdf=max(Frdf,0.15+0.85*(1.0-exp(-0.1/Sinb)));
610         Pardf=Par*Frdf;
611         Pdr=Par-Pardf;
612         Assim(Scp,Amax,Eff,Kdf,Lai,Sinb,Pdr,Pardf,&Fgros);
613         Dtgas=Dtgas+Fgros*wGauss[i];
614     }
615     *Dtga=Dtgas*Dayl*3600.0;
616 }

617 double limit(double ylow, double yhigh,double x)
618 {
619     double y;
620     if ((x >= ylow) && (x <=yhigh))
621         y = x;
622     else if (x < ylow)
623         y=ylow;
624     else if (x > yhigh)
625         y=yhigh;
626     return (y);
627 }

628 double max1(double a,double b, double c, double d)
629 {
630     double Maximo;
631     Maximo=max(a,b);
632     Maximo=max(Maximo,c);
633     Maximo=max(Maximo,d);
634     return (Maximo);
635 }

636 double NotNull(double x)
637 {
638     double y;
639     if (x==0.0)
640         y=1.0;
641     else
642         y=x;
643     return (y);
644 }

645 /*=====
646  * See sfuntmpl.doc for the optional S-function methods *
647  *=====*/

648 /*=====
649  * Required S-function trailer *

```

```
650  *=====*/
651  #ifdef MATLAB_MEX_FILE /* Is this file being compiled as a MEX-file? */
652  #include "simulink.c" /* MEX-file interface mechanism */
653  #else
654  #include "cg_sfun.h" /* Code generation registration function */
655  #endif
```

Apéndice E

Implementación C-Mex s-function del modelo LINTUL1 para la simulación del crecimiento potencial del cultivo de tomate de cáscara

```
1  /*
2  Irineo L. Lopez Cruz
3  Postgrado IAUIA, UACH
4  Chapingo Mex.
5  2007
6  */

7  #define S_FUNCTION_LEVEL 2
8  #define S_FUNCTION_NAME lintul1Tomate

9  /*
10 * Need to include simstruc.h for the definition of the SimStruct and
11 * its associated macro definitions.
12 */

13 #include "simstruc.h"
14 #include <math.h>
15 #include <stdlib.h>
16 #define ROWS 50
17 #define COLS 2

18 #define max(A,B) ((A) > (B) ? (A) : (B))
19 #define min(A,B) ((A) < (B) ? (A) : (B))
20 #define u(element)(*uPtrs[element])

21 /* Funciones default:FST*/
22 double minterp1(double,double,double,double,double);
23 double mafgen(double z[][COLS],double,int);
24 double limit(double, double,double );
25 double NotNull(double );
26 double max1(double ,double , double , double );
27 double insw(double ,double ,double );

28 /* Funciones del modelo LINTUL1 */
29 void gla(double ,double ,double ,double , double ,double ,double , double ,double , double ,double *);

30 /*Used to read the parameter number 1*/
31 #define x10 p[0]
32 #define x20 p[1]
33 #define x30 p[2]
34 #define x40 p[3]
35 #define x50 p[4]
36 #define x60 p[5]
37 #define x70 p[6]
38 #define LAII p[7]
39 #define SLA p[8]
40 #define TBASE p[9]
41 #define DAYEM p[10]
42 #define RGRL p[11]
43 #define TSUMAN p[12]
44 #define LAICR p[13]
45 #define RDRSHM p[14]
46 #define LUE p[15]
47 #define K p[16]
48 #define DeltaT p[17]
```

```

49  /* Error handling
50  * -----
51  *
52  * You should use the following technique to report errors encountered within
53  * an S-function:
54  *
55  *     ssSetErrorStatus(S,"Error encountered due to ...");
56  *     return;
57  *
58  * Note that the 2nd argument to ssSetErrorStatus must be persistent memory.
59  * It cannot be a local variable. For example the following will cause
60  * unpredictable errors:
61  *
62  *     mdlOutputs()
63  *     {
64  *         char msg[256];      {ILLEGAL: to fix use "static char msg[256];"}
65  *         sprintf(msg,"Error due to %s", string);
66  *         ssSetErrorStatus(S,msg);
67  *         return;
68  *     }
69  *
70  * See matlabroot/simulink/src/sfunctmpl.doc for more details.
71  */

72  /*=====
73  * S-function methods *
74  *=====*/

75  /* Function: mdlInitializeSizes =====
76  * Abstract:
77  * The sizes information is used by Simulink to determine the S-function
78  * block's characteristics (number of inputs, outputs, states, etc.).
79  */

80  static void mdlInitializeSizes(SimStruct *S)
81  {
82      /* See sfuntmpl.doc for more details on the macros below */

83      ssSetNumSFcnParams(S, 1); /* Number of expected parameters */
84      if (ssGetNumSFcnParams(S) != ssGetSFcnParamsCount(S)) {
85          /* Return if number of expected != number of actual parameters */
86          return;
87      }

88      ssSetNumContStates(S, 7);
89      ssSetNumDiscStates(S, 0);

90      if (!ssSetNumInputPorts(S, 1)) return;
91      /*Entradas del modelo*/
92      ssSetInputPortWidth(S, 0, 3);

93      ssSetInputPortDirectFeedThrough(S, 0, 1);

94      if (!ssSetNumOutputPorts(S, 1)) return;
95      ssSetOutputPortWidth(S, 0, 8);

96      ssSetNumSampleTimes(S, 1);
97      ssSetNumRWork(S, 0);
98      ssSetNumIWork(S, 0);
99      ssSetNumPWork(S, 0);
100      ssSetNumModes(S, 0);
101      ssSetNumNonsampledZCs(S, 0);

```

```

102     ssSetOptions(S, 0);
103 }

104 /* Function: mdlInitializeSampleTimes =====
105  * Abstract:
106  * This function is used to specify the sample time(s) for your
107  * S-function. You must register the same number of sample times as
108  * specified in ssSetNumSampleTimes.
109  */
110 static void mdlInitializeSampleTimes(SimStruct *S)
111 {
112     ssSetSampleTime(S, 0, CONTINUOUS_SAMPLE_TIME);
113     ssSetOffsetTime(S, 0, 0.0);
114 }
115 #define MDL_INITIALIZE_CONDITIONS /* Change to #undef to remove function */

116 #if defined(MDL_INITIALIZE_CONDITIONS)
117 /* Function: mdlInitializeConditions =====
118  * Abstract:
119  * In this function, you should initialize the continuous and discrete
120  * states for your S-function block. The initial states are placed
121  * in the state vector, ssGetContStates(S) or ssGetRealDiscStates(S).
122  * You can also perform any other initialization activities that your
123  * S-function may require. Note, this routine will be called at the
124  * start of simulation and if it is present in an enabled subsystem
125  * configured to reset states, it will be call when the enabled subsystem
126  * restarts execution to reset the states.
127  */
128 static void mdlInitializeConditions(SimStruct *S)
129 {
130     real_T *x0=ssGetContStates(S);
131     double *p=mxGetPr(ssGetSFcnParam(S,0));
132     x0[0]=x10;
133     x0[1]=x20;
134     x0[2]=x30;
135     x0[3]=x40;
136     x0[4]=x50;
137     x0[5]=x60;
138     x0[6]=x70;
139 }

140 #endif /* MDL_INITIALIZE_CONDITIONS */

141 #define MDL_START /* Change to #undef to remove function */
142 #if defined(MDL_START)
143 /* Function: mdlStart =====
144  * Abstract:
145  * This function is called once at start of model execution. If you
146  * have states that should be initialized once, this is the place
147  * to do it.
148  */
149 static void mdlStart(SimStruct *S)
150 {
151 }
152 #endif /* MDL_START */

153 /* Function: mdlOutputs =====
154  * Abstract:
155  * In this function, you compute the outputs of your S-function
156  * block. Generally outputs are placed in the output vector, ssGetY(S).
157  */

```

```

158 static void mdlOutputs(SimStruct *S, int_T tid)
159 {
160     real_T *y=ssGetOutputPortRealSignal(S,0);
161     real_T *x=ssGetContStates(S);
162     InputRealPtrsType uPtrs=ssGetInputPortRealSignalPtrs(S,0);
163     double *p=mxGetPr(ssGetSFcnParam(S,0));
164     real_T t=ssGetT(S);

165     /*InputRealPtrsType uPtrs=ssGetInputPortRealSignalPtrs(S,0);*/
166     y[0]=x[0];
167     y[1]=x[1];
168     y[2]=x[2];
169     y[3]=x[3];
170     y[4]=x[4];
171     y[5]=x[5];
172     y[6]=x[6];
173     y[7]=x[2]+x[3]+x[4]+x[5]+x[6];
174 }

174 #define MDL_UPDATE /* Change to #undef to remove function */
175 #if defined(MDL_UPDATE)
176     /* Function: mdlUpdate =====
177     * Abstract:
178     * This function is called once for every major integration time step.
179     * Discrete states are typically updated here, but this function is useful
180     * for performing any tasks that should only take place once per
181     * integration step.
182     */
183     static void mdlUpdate(SimStruct *S, int_T tid)
184     {
185     }
186 #endif /* MDL_UPDATE */

187 #define MDL_DERIVATIVES /* Change to #undef to remove function */
188 #if defined(MDL_DERIVATIVES)
189     /* Function: mdlDerivatives =====
190     * Abstract:
191     * In this function, you compute the S-function block's derivatives.
192     * The derivatives are placed in the derivative vector, ssGetdX(S).
193     */
194     static void mdlDerivatives(SimStruct *S)
195     {

196         double RdrT[5][2];
197         double FrtTb[9][2];
198         double FlvTb[7][2];
199         double FstTb[8][2];
200         double FsoTb[5][2];

201         real_T *dx=ssGetdX(S);
202         real_T *x=ssGetContStates(S);
203         InputRealPtrsType uPtrs=ssGetInputPortRealSignalPtrs(S,0);
204         double *p=mxGetPr(ssGetSFcnParam(S,0));
205         real_T t=ssGetT(S);

206         double Emergencia,DtMax,DtMin,Dtr,DavTemp,Dteff,RdrDv,RdrSh;
207         double Rdr,CrecLAI,MuerteLAI,Wlv, Flv,Fst,Fso,Frt;
208         double Glv,Gst, Gso,Grt,RLeafAreaIndex,Dlv;
209         double ParInt,GTotal,RWlvg,RtSum;

```

```

210 /*
211 Variables de entrada del modelo
212 u(0): Temperatura maxima
213 u(1): Temperatura minima
214 u(2): Radiacion solar

215 x[0]: Estado de desarrollo (DVS)
216 x[1]: IAF (IAF)
217 x[2]: biomasa de hojas verdes (wlvg)
218 x[3]: biomasa de hojas secas(wld)
219 x[4]: biomasa de raices(wrt)
220 x[5]: biomasa de tallos(wst)
221 x[6]: biomasa de organos(wso)
222 */

223 RdrT[0][0]=-10;
224 RdrT[1][0]=10;
225 RdrT[2][0]=15;
226 RdrT[3][0]=30;
227 RdrT[4][0]=50;
228 RdrT[0][1]=0.03;
229 RdrT[1][1]=0.03;
230 RdrT[2][1]=0.04;
231 RdrT[3][1]=0.09;
232 RdrT[4][1]=0.09;

233 FrtTb[0][0]=0;
234 FrtTb[1][0]=110;
235 FrtTb[2][0]=275;
236 FrtTb[3][0]=555;
237 FrtTb[4][0]=780;
238 FrtTb[5][0]=1055;
239 FrtTb[6][0]=1160;
240 FrtTb[7][0]=1305;
241 FrtTb[8][0]=2500;
242 FrtTb[0][1]=0.50;
243 FrtTb[1][1]=0.50;
244 FrtTb[2][1]=0.34;
245 FrtTb[3][1]=0.12;
246 FrtTb[4][1]=0.07;
247 FrtTb[5][1]=0.03;
248 FrtTb[6][1]=0.02;
249 FrtTb[7][1]=0.0;
250 FrtTb[8][1]=0.0;

251 FlvTb[0][0]=0.00;
252 FlvTb[1][0]=110;
253 FlvTb[2][0]=275;
254 FlvTb[3][0]=555;
255 FlvTb[4][0]=780;
256 FlvTb[5][0]=1055;
257 FlvTb[6][0]=2500;

258 FlvTb[0][1]=0.33;
259 FlvTb[1][1]=0.33;
260 FlvTb[2][1]=0.46;
261 FlvTb[3][1]=0.44;
262 FlvTb[4][1]=0.14;
263 FlvTb[5][1]=0.0;
264 FlvTb[6][1]=0.0;

265 FstTb[0][0]=0.0;

```

```

266 FstTb[1][0]=110;
267 FstTb[2][0]=275;
268 FstTb[3][0]=555;
269 FstTb[4][0]=780;
270 FstTb[5][0]=1055;
271 FstTb[6][0]=1160;
272 FstTb[7][0]=2500;

273 FstTb[0][1]=0.17;
274 FstTb[1][1]=0.17;
275 FstTb[2][1]=0.20;
276 FstTb[3][1]=0.44;
277 FstTb[4][1]=0.79;
278 FstTb[5][1]=0.97;
279 FstTb[6][1]=0.0;
280 FstTb[7][1]=0.0;

281 FsoTb[0][0]=0.0;
282 FsoTb[1][0]=1055;
283 FsoTb[2][0]=1160;
284 FsoTb[3][0]=1305;
285 FsoTb[4][0]=2500;
286 FsoTb[0][1]=0.0;
287 FsoTb[1][1]=0.0;
288 FsoTb[2][1]=0.98;
289 FsoTb[3][1]=1.0;
290 FsoTb[4][1]=1.0;

291 Emergencia=insw(t-DAYEM,0.0,1.0);
292 DtMax=u(0);
293 DtMin=u(1);
294 Dtr=u(2)/1000000.0;
295 Dtr=u(2);
296 DavTemp=0.5*(DtMax+DtMin);
297 Dteff=max(0.0,DavTemp-TBASE);

298 RdrSh=limit(0.0,RDRSHM,RDRSHM*(x[1]-LAICR)/LAICR);
299 RdrDv=insw((x[0]-TSUMAN),0.0,mafgn(Rdrt,DavTemp,5));
300 Rdr=max(RdrDv,RdrSh);
301 MuerteLAI=x[1]*Rdr;

302 Wlv=x[2]+x[3];

303 /*Particionamiento de la biomasa */
304 Flv=mafgn(FlvTb,x[0],7);
305 Fst=mafgn(FstTb,x[0],8);
306 Fso=mafgn(FsoTb,x[0],5);
307 Frt=mafgn(FrtTb,x[0],9);

308 /* Tasa de crecimiento total, tallos y raices*/

309 Dlv=x[2]*Rdr;
310 ParInt=0.5*Dtr*(1.0-exp(-K*x[1]));
311 GTot=LUE*ParInt;
312 Glv=Flv*GTot;

313 gla(t,DAYEM,Dteff,x[0],LAI,RGRL,DeltaT,SLA,x[1],Glv,&CrecLAI);

314 RLeafAreaIndex=CrecLAI-MuerteLAI;
315 RWlv=GTot*Flv-Dlv;

316 RtSum=Dteff*Emergencia;

```

```

317     Gst=GTotal*Fst;
318     Gso=GTotal*Fso;
319     Grt=GTotal*Frt;

320     /*TSum*/
321     dx[0]=RtSum;
322     /*LAI*/
323     dx[1]=RLeafAreaIndex;
324     /*WLVG*/
325     dx[2]=RWlvvg;
326     /*WLVD*/
327     dx[3]=Dlv;
328     /*WRT*/
329     dx[4]=Grt;
330     /*WST*/
331     dx[5]=Gst;
332     /*WSO*/
333     dx[6]=Gso;
334     }

335 #endif /* MDL_DERIVATIVES */

336 /* Function: mdlTerminate =====
337  * Abstract:
338  *   In this function, you should perform any actions that are necessary
339  *   at the termination of a simulation. For example, if memory was
340  *   allocated in mdlInitializeConditions, this is the place to free it.
341  */
342 static void mdlTerminate(SimStruct *S)
343 {
344 }

345 double mafgen(double z[][COLS],double x, int nrows)
346 {
347     int i;
348     double y;
349     for (i=0;i<nrows;i++){
350         if (z[i][0]==x){
351             y=z[i][1];
352             break;
353         }
354         else if ( ( z[i][0]< x) && (z[i+1][0]> x)){
355             y=minterp1(z[i][0],z[i][1],z[i+1][0],z[i+1][1],x);
356             break;
357         }
358     }
359     return(y);
360 }

361 double minterp1(double x1,double y1,double x2,double y2,double x)
362 {
363     double y;
364     y=y1+(y2-y1)*(x-x1)/(x2-x1);
365     return (y);
366 }

367 double insw(double x,double y1,double y2)
368 {
369     double y;
370     if (x<0)
371         y=y1;
372     else

```

```

373     y=y2;
374     return (y);
375 }

376 /* Subrutinas LINTUL1*/

377 void gla(double Dia,double Dayem,double DTeff,double TSum, double laii,double Rgrl,double Deltat,
378 double sla,double LAI, double Glv,double *CrecLai)

379 {
380     (*CrecLai)=sla*Glv;
381     if ((TSum < 330) && (LAI < 0.75))
382         (*CrecLai)=LAI*(exp(Rgrl*DTeff*Deltat)-1.0)/Deltat;

383     if ((Dia >=Dayem) && (LAI==0.0))
384         (*CrecLai=laii)/Deltat;
385     if (Dia < Dayem)
386         (*CrecLai)=0.0;
387 }

388 double limit(double ylow, double yhigh,double x)
389 {
390     double y;
391     if ((x >= ylow) && (x <=yhigh))
392         y = x;
393     else if (x < ylow)
394         y=ylow;
395     else if (x > yhigh)
396         y=yhigh;
397     return (y);
398 }

399 double max1(double a,double b, double c, double d)
400 {
401     double Maximo;
402     Maximo=max(a,b);
403     Maximo=max(Maximo,c);
404     Maximo=max(Maximo,d);
405     return (Maximo);
406 }

407 double NotNull(double x)
408 {
409     double y;
410     if (x==0.0)
411         y=1.0;
412     else
413         y=x;
414     return (y);
415 }

416 /*=====
417  * See sfuntmpl.doc for the optional S-function methods *
418  *=====*/

419 /*=====
420  * Required S-function trailer *
421  *=====*/

422 #ifdef MATLAB_MEX_FILE /* Is this file being compiled as a MEX-file? */
423 #include "simulink.c" /* MEX-file interface mechanism */
424 #else

```

```
425  #include "cg_sfuns.h"    /* Code generation registration function */
426  #endif
```

Apéndice F

Implementación C-Mex s-function del modelo SUCROS972 para la simulación del crecimiento limitado por agua del cultivo de tomate de cáscara

```
1  /*
2  Irineo L. Lopez Cruz
3  Postgrado IAUIA, UACH
4  Chapingo Mex.
5  2006
6  */

7  #define S_FUNCTION_LEVEL 2
8  #define S_FUNCTION_NAME sucros972

9  /*
10 * Need to include simstruc.h for the definition of the SimStruct and
11 * its associated macro definitions.
12 */

13 #include "simstruc.h"
14 #include <math.h>
15 #include <stdlib.h>
16 #define ROWS 50
17 #define COLS 2

18 #define max(A,B) ((A) > (B) ? (A) : (B))
19 #define min(A,B) ((A) < (B) ? (A) : (B))
20 #define u(element)(*uPtrs[element])

21 /* Funciones default:FST*/
22 double minterp1(double,double,double,double,double);
23 double mafgen(double z[][COLS],double,int);
24 double limit(double, double,double );
25 double NotNull(double );
26 double max1(double ,double , double , double );
27 double min1(double ,double , double );
28 double insw(double ,double ,double );

29 /* Funciones del modelo SUCROS */

30 void gla(double Time,double Doyem,double DTeff,double Dvs,double Rgrl,double Deltat,double Sla,double
31 Lai,double Glv,double *CrecLai);
32 void Astro(double Doy, double Lat, double *Sc,double *Ds0, double *Sinld, double *Cosld, double *Dayl,
33 double *Dsinb, double *Dsinbe);
34 void Assim(double Scp, double Amax,double Eff,double Kdf, double Lai, double Sinb, double Pardr, double
35 Pardf,double *Fgros);
36 void Totass(double Day, double Lat, double Dtr, double Scp, double Amax, double Eff, double Kdf, double
37 Lai, double *Dayl,double *Dtga, double *Ds0);
38 void SubGrt(double ,double ,double ,double ,double ,double , double , double , double ,double ,double ,double ,
39 double ,double ,double ,double ,double *);
40 void SubFr(double Wcl,double Wcfc,double P, double Wcwp, double Wcwet,double Wcst, double *Wse);

41 /*Used to read the parameter number 1*/
42 #define x10 p[0]
43 #define x20 p[1]
44 #define x30 p[2]
45 #define x40 p[3]
46 #define x50 p[4]
47 #define x60 p[5]
```

48	#define x70	p[6]
49	#define x80	p[7]
50	#define x90	p[8]
51	#define x100	p[9]
52	#define x110	p[10]
53	#define x120	p[11]
54	#define x130	p[12]
55	#define x140	p[13]
56	#define x150	p[14]
57	#define x160	p[15]
58	#define x170	p[16]
59	#define x180	p[17]
60	#define x190	p[18]
61	#define x200	p[19]
62	#define x210	p[20]
63	#define x220	p[21]
64	#define x230	p[22]
65	#define x240	p[23]
66	#define x250	p[24]
67	#define x260	p[25]
68	#define LATT	p[26]
69	#define DAYEM	p[27]
70	#define RGRL	p[28]
71	#define TBASE	p[29]
72	#define SLA	p[30]
73	#define AMX	p[31]
74	#define EFF	p[32]
75	#define KDF	p[33]
76	#define SCP	p[34]
77	#define Q10	p[35]
78	#define TREF	p[36]
79	#define MAINLV	p[37]
80	#define MAINST	p[38]
81	#define MAINRT	p[39]
82	#define MAINSO	p[40]
83	#define ASRQLV	p[41]
84	#define ASRQST	p[42]
85	#define ASRQRT	p[43]
86	#define ASRQSO	p[44]
87	#define DeltaT	p[45]
88	#define LAICR	p[46]
89	#define FRTRL	p[47]
90	#define CFST	p[48]
91	#define CFLV	p[49]
92	#define CFRT	p[50]
93	#define CFSO	p[51]
94	#define FRDR	p[52]
95	#define CONVL	p[53]
96	#define LHVAP	p[54]
97	#define PSYCH	p[55]
98	#define BOLTZM	p[56]
99	#define WCST1	p[57]
100	#define WCST2	p[58]
101	#define WCST3	p[59]
102	#define WCST4	p[60]
103	#define WCFC1	p[61]
104	#define WCFC2	p[62]
105	#define WCFC3	p[63]
106	#define WCFC4	p[64]
107	#define A1	p[65]
108	#define B1	p[66]
109	#define TKL1	p[67]

```

110 #define TKL2          p[68]
111 #define TKL3          p[69]
112 #define TKL4          p[70]
113 #define WCWP1         p[71]
114 #define WCWP2         p[72]
115 #define WCWP3         p[73]
116 #define WCWP4         p[74]
117 #define WCWET1        p[75]
118 #define WCWET2        p[76]
119 #define WCWET3        p[77]
120 #define WCWET4        p[78]
121 #define INTC          p[79]
122 #define MDRATE        p[80]
123 #define WCAD1         p[81]
124 #define WCAD2         p[82]
125 #define WCAD3         p[83]
126 #define WCAD4         p[84]
127 #define EES           p[85]
128 #define TRANSC        p[86]
129 #define EZRTC         p[87]
130 #define ZRTMC         p[88]
131 #define ZRTMS         p[89]
132 #define WCUMI         p[90]

133 /* Error handling
134  * -----
135  *
136  * You should use the following technique to report errors encountered within
137  * an S-function:
138  *
139  *     ssSetErrorStatus(S,"Error encountered due to ...");
140  *     return;
141  *
142  * Note that the 2nd argument to ssSetErrorStatus must be persistent memory.
143  * It cannot be a local variable. For example the following will cause
144  * unpredictable errors:
145  *
146  *     mdlOutputs()
147  *     {
148  *         char msg[256];      {ILLEGAL: to fix use "static char msg[256];"}
149  *         sprintf(msg,"Error due to %s", string);
150  *         ssSetErrorStatus(S,msg);
151  *         return;
152  *     }
153  *
154  * See matlabroot/simulink/src/sfunctmpl.doc for more details.
155  */

156 /*=====
157  * S-function methods *
158  *=====*/

159 /* Function: mdlInitializeSizes =====
160  * Abstract:
161  *   The sizes information is used by Simulink to determine the S-function
162  *   block's characteristics (number of inputs, outputs, states, etc.).
163  */
164 static void mdlInitializeSizes(SimStruct *S)
165 {
166     /* See sfuntmpl.doc for more details on the macros below */

167     ssSetNumSFcnParams(S, 1); /* Number of expected parameters */

```

```

168     if (ssGetNumSFcnParams(S) != ssGetSFcnParamsCount(S)) {
169         /* Return if number of expected != number of actual parameters */
170         return;
171     }

172     ssSetNumContStates(S, 26);
173     ssSetNumDiscStates(S, 0);

174     if (!ssSetNumInputPorts(S, 1)) return;
175     /*Entradas del modelo*/
176     ssSetInputPortWidth(S, 0, 6);

177     ssSetInputPortDirectFeedThrough(S, 0, 0);

178     if (!ssSetNumOutputPorts(S, 1)) return;
179     ssSetOutputPortWidth(S, 0, 29);

180     ssSetNumSampleTimes(S, 1);
181     ssSetNumRWork(S, 0);
182     ssSetNumIWork(S, 0);
183     ssSetNumPWork(S, 0);
184     ssSetNumModes(S, 0);
185     ssSetNumNonsampledZCs(S, 0);

186     ssSetOptions(S, 0);
187 }

188 /* Function: mdlInitializeSampleTimes =====
189 * Abstract:
190 * This function is used to specify the sample time(s) for your
191 * S-function. You must register the same number of sample times as
192 * specified in ssSetNumSampleTimes.
193 */
194 static void mdlInitializeSampleTimes(SimStruct *S)
195 {
196     ssSetSampleTime(S, 0, CONTINUOUS_SAMPLE_TIME);
197     ssSetOffsetTime(S, 0, 0.0);
198 }
199

200 #define MDL_INITIALIZE_CONDITIONS /* Change to #undef to remove function */
201 #if defined(MDL_INITIALIZE_CONDITIONS)
202 /* Function: mdlInitializeConditions =====
203 * Abstract:
204 * In this function, you should initialize the continuous and discrete
205 * states for your S-function block. The initial states are placed
206 * in the state vector, ssGetContStates(S) or ssGetRealDiscStates(S).
207 * You can also perform any other initialization activities that your
208 * S-function may require. Note, this routine will be called at the
209 * start of simulation and if it is present in an enabled subsystem
210 * configured to reset states, it will be call when the enabled subsystem
211 * restarts execution to reset the states.
212 */
213 static void mdlInitializeConditions(SimStruct *S)
214 {
215     real_T *x0=ssGetContStates(S);
216     double *p=mxGetPr(ssGetSFcnParam(S,0));
217     x0[0]=x10;
218     x0[1]=x20;
219     x0[2]=x30;
220     x0[3]=x40;
221     x0[4]=x50;

```

```

222         x0[5]=x60;
223         x0[6]=x70;
224         x0[7]=x80;
225         x0[8]=x90;
226         x0[9]=x100;
227         x0[10]=x110;
228         x0[11]=x120;
229         x0[12]=x130;
230         x0[13]=x140;
231         x0[14]=x150;
232         x0[15]=x160;
233         x0[16]=x170;
234         x0[17]=x180;
235         x0[18]=x190;
236         x0[19]=x200;
237         x0[20]=x210;
238         x0[21]=x220;
239         x0[22]=x230;
240         x0[23]=x240;
241         x0[24]=x250;
242         x0[25]=x260;
243     }
244 #endif /* MDL_INITIALIZE_CONDITIONS */

245 #define MDL_START /* Change to #undef to remove function */
246 #if defined(MDL_START)
247     /* Function: mdlStart =====
248     * Abstract:
249     *   This function is called once at start of model execution. If you
250     *   have states that should be initialized once, this is the place
251     *   to do it.
252     */
253     static void mdlStart(SimStruct *S)
254     {
255     }
256 #endif /* MDL_START */

257 /* Function: mdlOutputs =====
258 * Abstract:
259 *   In this function, you compute the outputs of your S-function
260 *   block. Generally outputs are placed in the output vector, ssGetY(S).
261 */
262 static void mdlOutputs(SimStruct *S, int_T tid)
263 {

264     double DvrVt[3][2];
265     double DvrRt[3][2];
266     double FlvTb[7][2];
267     double FshTb[14][2];
268     double Fsttb[8][2];
269     double Amdvst[4][2];
270     double Amtmpt[6][2];

271     real_T *y=ssGetOutputPortRealSignal(S,0);
272     real_T *x=ssGetContStates(S);
273     InputRealPtrsType uPtrs=ssGetInputPortRealSignalPtrs(S,0);
274     double *p=mxGetPr(ssGetSFcnParam(S,0));
275     real_T t=ssGetT(S);

276     double DevRate,Emergencia,Dia, DtMax,DtMin,Dtr,DavTemp,RLeafAreaIndex;
277     double CrecLAI,MuerteLAI,GlV,Gsh,Gst, Gso,Grt, Rdr,RdrDv,RdrSh,RdrLt,RdrSl;
278     double Fsh,Flv,Fst,Fso,Frt,Gtw, GPhot,Maint,Asrq,Dtga,Dso, Maints,Teff,Mndvs;

```

```

279 double Wlv,RDiv, RWlv, Amax,Amdvs,Amtmp,DdTm,Dteff;
280 double Dayl, Transl;
281 double Tnassi,Chkin,Chkfl, Rtnass,CO2Rt,CO2Lv,CO2St, CO2So,Chkdif;

282 /*
283 x[0] : Dvs
284 x[1] : Lai
285 x[2] : wlv
286 x[3] : wlv
287 x[4] : wrt
288 x[5] : wst
289 x[6] : wso
290 x[7] : Tnass
291 x[8] : Train
292 x[9] : TPenm
293 x[10]: Tevapr
294 x[11]: Tevapr
295 x[12]: wl1
296 x[13]: wl2
297 x[14]: wl3
298 x[15]: wl4
299 x[16]: Tdrain
300 x[17]: Tainc
301 x[18]: Trnoff
302 x[19]: zrt
303 x[20]: Tptran
304 x[21]: Tattran
305 x[22]: Dslr
306 x[23]: Tpevap
307 x[24]: Taevap
308 x[25]: Tdtga
309 */

310 y[0]=x[0];
311 y[1]=x[1];
312 y[2]=x[2];
313 y[3]=x[3];
314 y[4]=x[4];
315 y[5]=x[5];
316 y[6]=x[6];
317 /* biomasa foliar*/
318 y[7]=x[2]+x[3];
319 /*biomasa de la parte aerea */
320 y[8]=x[2]+x[3]+x[5]+x[6];
321 /*biomasa total incluyendo raices*/
322 y[9]=x[2]+x[3]+x[4]+x[5]+x[6];
323 y[10]=x[7];
324 y[11]=x[8];
325 y[12]=x[9];
326 y[13]=x[10];
327 y[14]=x[11];
328 y[15]=x[12];
329 y[16]=x[13];
330 y[17]=x[14];
331 y[18]=x[15];
332 y[19]=x[16];
333 y[20]=x[17];
334 y[21]=x[18];
335 y[22]=x[19];
336 y[23]=x[20];
337 y[24]=x[21];
338 y[25]=x[22];

```

```

339 y[26]=x[23];
340 y[27]=x[24];
341 y[28]=x[25];
342 }

343 #define MDL_UPDATE /* Change to #undef to remove function */
344 #if defined(MDL_UPDATE)
345 /* Function: mdlUpdate =====
346 * Abstract:
347 * This function is called once for every major integration time step.
348 * Discrete states are typically updated here, but this function is useful
349 * for performing any tasks that should only take place once per
350 * integration step.
351 */
352 static void mdlUpdate(SimStruct *S, int_T tid)
353 {
354 }
355 #endif /* MDL_UPDATE */

356 #define MDL_DERIVATIVES /* Change to #undef to remove function */
357 #if defined(MDL_DERIVATIVES)
358 /* Function: mdlDerivatives =====
359 * Abstract:
360 * In this function, you compute the S-function block's derivatives.
361 * The derivatives are placed in the derivative vector, ssGetdX(S).
362 */
363 static void mdlDerivatives(SimStruct *S)
364 {
365 double DvrVt[3][2];
366 double DvrRt[3][2];
367 double FlvTb[7][2];
368 double FshTb[14][2];
369 double Fsttb[8][2];
370 double Amdvst[4][2];
371 double Amtmpt[6][2];
372 double FsoTb[4][2];
373 double EdptfTb[7][2];

374 real_T *dx=ssGetdX(S);
375 real_T *x=ssGetContStates(S);
376 InputRealPtrsType uPtrs=ssGetInputPortRealSignalPtrs(S,0);
377 double *p=mxGetPr(ssGetSFcnParam(S,0));
378 real_T t=ssGetT(S);

379 double DevRate,Emergencia,Dia, DtMax,DtMin,Dtr,DavTemp,RLeafAreaIndex;
380 double CrecLAI,MuerteLAI,GlV,Gsh,Gst, Gso,Grt, Rdr,RdrDv,RdrSh,RdrLt,RdrSl;
381 double Fsh,Flv,Fst,Fso,Frt,Gtw, GPhot,Maint,Asrq,Dtga, Maints,Teff,Mndvs;
382 double Wlv,RDlv, RWlvG, Amax,Amdvs,Amtmp,DdTmp,Dteff;
383 double Transl,Dlv,Ds0, LongDia;
384 double Tnassi,Chkin,Chkfl, Rtnass,CO2Rt,CO2Lv,CO2St, CO2So,Chkdif;
385 int i;
386 double Svp,Slope,Bbrad,Clear, Fclear,Avp,Fvap,Rlwn, Wcl1, Wcl2, Wcl3,Wcl4;
387 double Wsert,Albs,Alb,Nrad, Evapr,Penman,Evapd;
388 double TKlt,Zrtm,Wds, Wdf, Dryp, Aintc, RainOff, WIFI1, WIFI2, WIFI3, WIFI4, WIFI5;
389 double PeVap,Evsh,Evsd,AeVap, FeVl1, FeVl2, FeVl3, FeVl4, FeVlt;
390 double EvsW1,EvsW2,EvsW3,EvsW4, Ptrans, P,Wse1, Wse2,Wse3,Wse4, Erlb, Zrt1,Zrt2,Zrt3,Zrt4;
391 double RwCl1,RwCl2,RwCl3,RwCl4, Trrm, TrWl1,TrWl2,TrWl3,TrWl4,Atrans;
392 double Ezrt,Rdslr,Tar,Cropf,PCew,CPew,Check,WCum, Fshp;
393 /* Tabla de tasa de desarrollo */
394 DvrVt[0][0]=-10.0;
395 DvrVt[1][0]=0;
396 DvrVt[2][0]=30;

```

```

397     DvrVt[0][1]=0;
398     DvrVt[1][1]=0;
399     DvrVt[2][1]=0.027;

400     DvrRt[0][0]=-10.0;
401     DvrRt[1][0]=0;
402     DvrRt[2][0]=30;
403     DvrRt[0][1]=0;
404     DvrRt[1][1]=0;
405     DvrRt[2][1]=0.031;
406     /* Tablas de division de la biomasa en diferentes organos */
407     FlvTb[0][0]=0.00;
408     FlvTb[1][0]=0.10;
409     FlvTb[2][0]=0.25;
410     FlvTb[3][0]=0.50;
411     FlvTb[4][0]=0.70;
412     FlvTb[5][0]=0.95;
413     FlvTb[6][0]=2.50;

414     FlvTb[0][1]=0.65;
415     FlvTb[1][1]=0.65;
416     FlvTb[2][1]=0.70;
417     FlvTb[3][1]=0.50;
418     FlvTb[4][1]=0.15;
419     FlvTb[5][1]=0.00;
420     FlvTb[6][1]=0.00;

421     FshTb[0][0]=0.00;
422     FshTb[1][0]=0.10;
423     FshTb[2][0]=0.20;
424     FshTb[3][0]=0.35;
425     FshTb[4][0]=0.40;
426     FshTb[5][0]=0.50;
427     FshTb[6][0]=0.60;
428     FshTb[7][0]=0.70;
429     FshTb[8][0]=0.80;
430     FshTb[9][0]=0.90;
431     FshTb[10][0]=1.00;
432     FshTb[11][0]=1.10;
433     FshTb[12][0]=1.20;
434     FshTb[13][0]=2.50;

435     FshTb[0][1]=0.50;
436     FshTb[1][1]=0.50;
437     FshTb[2][1]=0.60;
438     FshTb[3][1]=0.78;
439     FshTb[4][1]=0.83;
440     FshTb[5][1]=0.87;
441     FshTb[6][1]=0.90;
442     FshTb[7][1]=0.93;
443     FshTb[8][1]=0.95;
444     FshTb[9][1]=0.97;
445     FshTb[10][1]=0.98;
446     FshTb[11][1]=0.99;
447     FshTb[12][1]=1.0;
448     FshTb[13][1]=1.0;

449     Fsttb[0][0]=0.00;
450     Fsttb[1][0]=0.10;
451     Fsttb[2][0]=0.25;
452     Fsttb[3][0]=0.50;
453     Fsttb[4][0]=0.70;

```

```

454     Fsttb[5][0]=0.95;
455     Fsttb[6][0]=1.05;
456     Fsttb[7][0]=2.50;

457     Fsttb[0][1]=0.35;
458     Fsttb[1][1]=0.35;
459     Fsttb[2][1]=0.30;
460     Fsttb[3][1]=0.50;
461     Fsttb[4][1]=0.85;
462     Fsttb[5][1]=1.00;
463     Fsttb[6][1]=0.00;
464     Fsttb[7][1]=0.00;

465     FsoTb[0][0]=0.00;
466     FsoTb[1][0]=0.95;
467     FsoTb[2][0]=1.05;
468     FsoTb[3][0]=2.50;

469     FsoTb[0][1]=0.00;
470     FsoTb[1][1]=0.00;
471     FsoTb[2][1]=1.00;
472     FsoTb[3][1]=1.00;

473     /*Tablas de asimilacion de CO2 */
474     Amdvst[0][0]=0.0;
475     Amdvst[1][0]=1.0;
476     Amdvst[2][0]=2.0;
477     Amdvst[3][0]=2.5;

478     Amdvst[0][1]=1.0;
479     Amdvst[1][1]=1.0;
480     Amdvst[2][1]=0.5;
481     Amdvst[3][1]=0.0;

482     Amtmpt[0][0]=-10.00;
483     Amtmpt[1][0]=0.00;
484     Amtmpt[2][0]=10.00;
485     Amtmpt[3][0]=25.00;
486     Amtmpt[4][0]=35.00;
487     Amtmpt[5][0]=50.00;

488     Amtmpt[0][1]=0.00;
489     Amtmpt[1][1]=0.00;
490     Amtmpt[2][1]=1.00;
491     Amtmpt[3][1]=1.00;
492     Amtmpt[4][1]=0.00;
493     Amtmpt[5][1]=0.00;

494     EdptfTb[0][0]=-0.5;
495     EdptfTb[1][0]=-0.05;
496     EdptfTb[2][0]=0.0;
497     EdptfTb[3][0]=0.15;
498     EdptfTb[4][0]=0.3;
499     EdptfTb[5][0]=0.5;
500     EdptfTb[6][0]=2.0;

501     EdptfTb[0][1]=0.0;
502     EdptfTb[1][1]=0.0;
503     EdptfTb[2][1]=0.15;
504     EdptfTb[3][1]=0.6;
505     EdptfTb[4][1]=0.8;
506     EdptfTb[5][1]=1.0;

```

```

507     EdptfTb[6][1]=1.0;

508     /* Dia = 1.0+fmod(t-1,365.0);*/
509     Dia =fmod(t,365.0);
510     Emergencia=insw(t-DAYEM,0.0,1.0);

511     DtMax=u(0);
512     DtMin=u(1);
513     Dtr=u(2)*1000000.0;
514     DavTemp=0.5*(DtMax+DtMin);
515     DevRate=insw(x[0]-1.0,mafgen(DvrVt,DavTemp,3),mafgen(DvrRt,DavTemp,3))*Emergencia;
516     RdrSh=limit(0.0,0.03,0.03*(x[1]-LAICR)/LAICR);
517     RdrLt=insw(x[0]-1.25,0.0,limit(0.0,1.0,(6-DavTemp)/6));
518     RdrSl=insw(x[0]-1.35,0.0005,0.0030);
519     RdrDv=insw(x[0]-1.0,0,DevRate/(max(0.1,2.0-x[0]))*FRDR);
520     /*Rdr=insw(x[0]-1.0,0.0,max1(RdrDv,RdrSh,RdrLt,0.001));*/
521     Rdr=max(RdrDv,RdrSh);
522     MuerteLAI=x[1]*Rdr;
523     /* Tasa de fotosintesis a saturacion de luz */

524     DdTmp=DtMax-0.25*(DtMax-DtMin);
525     Amdvs=mafgen(Amdvst,x[0],4);
526     Amtmp=mafgen(Amtmpt,DdTmp,6);
527     Amax=AMX*Amdvs*Amtmp;
528     Totass(Dia,LATT,Dtr,SCP,Amax,EFF,KDF,x[1],&LongDia,&Dtga,&Ds0);

529     /* Variables auxiliares */
530     TKlt=TKL1+TKL2+TKL3+TKL4;
531     Zrtm=min1(ZRTMC,ZRTMS,TKlt);

532     /* Penman Monteith equation*/

533     Svp=0.611*exp(17.4*DavTemp/(DavTemp+239.0));
534     Slope=4158.6*Svp/pow((DavTemp+239.0),2.0);
535     Bbrad=BOLTZM*pow((DavTemp+273.0),4.0);
536     Clear=limit(0.0,1.0,((Dtr/Ds0)-A1)/B1);
537     Fclear=0.1+0.9*Clear;
538     Avp=u(4);
539     Wds=u(5);
540     Fvap=0.56*0.079*sqrt(Avp*10);
541     Rlwn=Bbrad*Fvap*Fclear*86400.0;
542     Wcl1=x[12]/TKL1;
543     Wcl2=x[13]/TKL2;
544     Wcl3=x[14]/TKL3;
545     Wcl4=x[15]/TKL4;

    SubGrt(x[19],Zrtm,x[0],TKL1,TKL2,TKL3,TKL4,Wcl1,Wcl2,Wcl3,Wcl4,WCWP1,WCWP2,WCWP3,WCWP4,
    &Wsert);

546     Albs=0.25*(1.0-0.5*Wcl1/WCST1);
547     Alb=Albs*exp(-0.5*x[1])+0.25*(1.0-exp(-0.5*x[1]));
548     Nrad=(1.0-Alb)*Dtr-Rlwn;
549     Evapr=(1.0/LHVP)*(Slope/(Slope+PSYCH))*Nrad;
550     Wdf=2.63*(1.0+0.54*Wds)*PSYCH;
551     Dryp=(Svp-Avp)*Wdf;
552     Evapd=Dryp/(Slope+PSYCH);
553     Penman=Evapr+Evapd;

554     Aintc=min(u(3),INTC*x[1]);
555     RainOff=max(0.15*(u(3)-Aintc-10.0),u(3)-Aintc-(WCST1*TKL1-x[12])/(2.0*DeltaT));
556     WIFI1=u(3)-Aintc-RainOff;
557     WIFI2=max(0.0,min(x[12]-WCFC1*TKL1,WCST2*TKL2-x[13])/(2*DeltaT));
558     WIFI3=max(0.0,min(x[13]-WCFC2*TKL2,WCST3*TKL3-x[14])/(2*DeltaT));

```

```

559   WIFl4=max(0.0,min(x[14]-WCFC3*TKL3,WCST4*TKL4-x[15])/(2*DeltaT));
560   WIFl5=max(0.0,min((x[15]-WCFC4*TKL4)/(2*DeltaT),MDRATE));

561   /* Evaporacion */
562   PeVap=exp(-0.5*x[1])*(Evapr+Evapd);
563   Evsh=min(PeVap,(x[12]-WCAD1*TKL1)/DeltaT+WIFl1);
564   Evsd=min(PeVap,0.6*PeVap*(sqrt(x[22]+1.0)-sqrt(x[22]))+WIFl1);
565   AeVap=insw(u(3)-0.5,Evsd,Evsh);

566   FeVl1=max(x[12]-WCAD1*TKL1,0.1)*exp(-EES*(0.5*TKL1));
567   FeVl2=max(x[13]-WCAD2*TKL2,0.1)*exp(-EES*(TKL1+0.5*TKL2));
568   FeVl3=max(x[14]-WCAD3*TKL3,0.1)*exp(-EES*(TKL1+TKL2+0.5*TKL3));
569   FeVl4=max(x[15]-WCAD4*TKL4,0.1)*exp(-EES*(TKL1+TKL2+TKL3+0.5*TKL4));
570   FeVlt=FeVl1+FeVl2+FeVl3+FeVl4;

571   EvsW1=AeVap*(FeVl1/FeVlt);
572   EvsW2=AeVap*(FeVl2/FeVlt);
573   EvsW3=AeVap*(FeVl3/FeVlt);
574   EvsW4=AeVap*(FeVl4/FeVlt);

575   /*Transpiracion*/

576   Ptrans=(1.0-exp(-0.5*x[1]))*Evapr+Evapd*min(2.0,x[1])-0.5*Aintc;
577   P=TRANSC/(TRANSC+Ptrans);

578   SubFr(Wcl1,WCFC1,P,WCWP1,WCWET1,WCST1,&Wse1);
579   SubFr(Wcl2,WCFC2,P,WCWP2,WCWET2,WCST2,&Wse2);
580   SubFr(Wcl3,WCFC3,P,WCWP3,WCWET3,WCST3,&Wse3);
581   SubFr(Wcl4,WCFC4,P,WCWP4,WCWET4,WCST4,&Wse4);
582   Zrt1=limit(0.0,TKL1,x[19]);
583   Zrt2=limit(0.0,TKL2,x[19]-TKL1);
584   Zrt3=limit(0.0,TKL3,x[19]-TKL1-TKL2);
585   Zrt4=limit(0.0,TKL4,x[19]-TKL1-TKL2-TKL3);
586   RwcI1=(Wcl1-WCWP1)/(WCFC1-WCWP1);
587   RwcI2=(Wcl2-WCWP2)/(WCFC2-WCWP2);
588   RwcI3=(Wcl3-WCWP3)/(WCFC3-WCWP3);
589   RwcI4=(Wcl4-WCWP4)/(WCFC4-WCWP4);

590   Erlb=Zrt1*mafgen(EdptfTb,RwcI1,7)+Zrt2*mafgen(EdptfTb,RwcI2,7)+Zrt3*mafgen(EdptfTb,RwcI3,7)+Zrt4*
591   mafgen(EdptfTb,RwcI4,7);

592   Trrm=Ptrans/NotNull(Erlb);

593   TrWl1=Trrm*Wse1*Zrt1*mafgen(EdptfTb,RwcI1,7);
594   TrWl2=Trrm*Wse2*Zrt2*mafgen(EdptfTb,RwcI2,7);
595   TrWl3=Trrm*Wse3*Zrt3*mafgen(EdptfTb,RwcI3,7);
596   TrWl4=Trrm*Wse4*Zrt4*mafgen(EdptfTb,RwcI4,7);
597   Atrans=TrWl1+TrWl2+TrWl3+TrWl4;
598   Ezrt=EZRTC*Wsert*Amtp;
599   RdsI=insw(u(3)-0.5,1.0,-(x[22]-1.0)/DeltaT);
600   Tar=x[21]*1000.0/NotNull(x[25]);
601   Cropf=(Ptrans+PeVap)/Penman;
602   PCew=Atrans/NotNull(Ptrans);
603   CPew=min(1.0,0.5+Atrans/NotNull(Ptrans));
604   WCum=x[12]+x[13]+x[14]+x[15];

605   Check=x[8]+WCUMI-x[17]-x[18]-x[16]-WCum-x[21]-x[11];

606   GPhot=Dtga*PCew*30.0/44.0;

607   Wlv=x[2]+x[3];
608   Mndvs=x[2]/(NotNull(Wlv));

```

```

609     Teff=pow(Q10,(DavTemp-TREF)/10.0);

610     Maints=MAINLV*x[2]+MAINST*x[5]+MAINRT*x[4]+MAINSO*x[6];
611     /*Maint=min(GPhot,Maints*Teff*Mndvs);*/
612     Maint=Maints*Teff*Mndvs*Emergencia;

613     /*Particionamiento de la biomasa */
614     Fshp=mafgen(FshTb,x[0],14);
615     Fsh=(Fshp*CPew)/(1.0+(CPew-1.0)*Fshp);

616     Flv=mafgen(FlvTb,x[0],7);
617     Fst=mafgen(Fsttb,x[0],8);
618     Fso=mafgen(FsoTb,x[0],4);

619     /*Fso=1.0-Flv-Fst;*/

620     Frt=1.0-Fsh;
621     Gso=Fso*Fsh*Gtw;
622     Asrq=Fsh*(ASRQLV*Flv+ASRQST*Fst+ASRQSO*Fso)+ASRQRT*Frt;
623     Gtw=(GPhot-Maint+CONVL*Transl*CFST*30.0/12.0)/Asrq;
624     /* Tasa de crecimiento total, tallos y raices*/
625     Gsh=Fsh*Gtw;
626     Glv=Flv*Fsh*Gtw;
627     Gst=Fst*Fsh*Gtw-Transl;
628     Gso=Fso*Fsh*Gtw;
629     Grt=Frt*Gtw;
630     Dteff=max(0.0,DavTemp-TBASE);
631     Transl = insw(x[0]-1.0,0.0,x[5]*DevRate*FRTRL);

632     gla(t,DAYEM,Dteff,x[0],RGRL,DeltaT,SLA,x[1],Glv,&CrecLAI);

633     RLeafAreaIndex=CrecLAI-MuerteLAI;
634     Dlv=x[2]*MuerteLAI/(NotNull(x[1]));
635     RWlvG=Glv-Dlv;

636     Chkin=Wlv*CFLV+x[5]*CFST+x[4]*CFRT+x[6]*CFSO;
637     Chkfl=x[7]*(12/44.0);
638     CO2Rt=44.0/12.0*(ASRQRT*12.0/30.0-CFRT);
639     CO2Lv=44.0/12.0*(ASRQLV*12.0/30.0-CFLV);
640     CO2St=44.0/12.0*(ASRQST*12.0/30.0-CFST);
641     CO2So=44.0/12.0*(ASRQSO*12.0/30.0-CFSO);
642     Chkdif=(Chkin-Chkfl)/NotNull(Chkin);
643     Rtnass=((GPhot-Maint)*44.0/30.0)-(Grt*CO2Rt+Glv*CO2Lv+(Gst+Transl)*CO2St+Gso*CO2So+(1.0-
644     CONVL)*Transl*CFST*44.0/12.0);

645     /*
646     Variables de entrada del modelo
647     u(1): Temperatura minima (°C)
648     u(0): Temperatura maxima (°C)
649     u(2): Radiacion solar (J /m2/d)
650     u(3) : lluvia (mm/d)
651     u(4) : Presion de vapor (kPa)
652     u(5) : Velocidad del viento (m/s)
653     x[0] : Dvs
654     x[1] : Lai
655     x[2] : wlvG
656     x[3] : wlvD
657     x[4] : wrt
658     x[5] : wst
659     x[6] : wso
660     x[7] : Tnass
661     x[8] : Train

```

```

662     x[9] : TPenm
663     x[10]: Tevapr
664     x[11]: Tevapd
665     x[12]: wl1
666     x[13]: wl2
667     x[14] :wl3
668     x[15] : wl4
669     x[16] : Tdrain
670     x[17] : Tainc
671     x[18] : Trnoff
672     x[19] : zrt
673     x[20] : Tptran
674     x[21] : Tattran
675     x[22] : Dslr
676     x[23] : Tpevap
677     x[24] : Taevap
678     x[25] : Tdtga
679 */

680     dx[0]=DevRate;
681     dx[1]=RLeafAreaIndex;
682     dx[2]=RWlvg;
683     dx[3]=Dlv;
684     dx[4]=Grt;
685     dx[5]=Gst;
686     dx[6]=Gso;
687     dx[7]=Rtnass;
688     dx[8]=u(3);
689     dx[9]=Penman;
690     dx[10]=Evapr;
691     dx[11]=Evapd;
692     dx[12]=WIFI1-WIFI2-EvsW1-TrWI1;
693     dx[13]=WIFI2-WIFI3-EvsW2-TrWI2;
694     dx[14]=WIFI3-WIFI4-EvsW3-TrWI3;
695     dx[15]=WIFI4-WIFI5-EvsW4-TrWI4;
696     dx[16]=WIFI5;
697     dx[17]=Aintc;
698     dx[18]=RainOff;
699     dx[19]=Ezrt;
700     dx[20]=Ptrans;
701     dx[21]=Atrans;
702     dx[22]=Rdslr;
703     dx[23]=PeVap;
704     dx[24]=AeVap;
705     dx[25]=Dtga;
706     }

707 #endif /* MDL_DERIVATIVES */

708 /* Function: mdlTerminate =====
709  * Abstract:
710  *   In this function, you should perform any actions that are necessary
711  *   at the termination of a simulation. For example, if memory was
712  *   allocated in mdlInitializeConditions, this is the place to free it.
713  */
714 static void mdlTerminate(SimStruct *S)
715 {
716 }

717 double mafgen(double z[][COLS],double x, int nrows)
718 {
719     int i;

```

```

720 double y;

721 for (i=0;i<nrows;i++){
722     if (z[i][0]==x){
723         y=z[i][1];
724         break;
725     }
726     else if ( (z[i][0]<x) && (z[i+1][0]>x)){
727         y=minterp1(z[i][0],z[i][1],z[i+1][0],z[i+1][1],x);
728         break;
729     }
730 }
731 return(y);
732 }

733 double minterp1(double x1,double y1,double x2,double y2,double x)
734 {
735     double y;
736     y=y1+(y2-y1)*(x-x1)/(x2-x1);
737     return (y);
738 }

739 double insw(double x,double y1,double y2)
740 {
741     double y;
742     if (x<0)
743         y=y1;
744     else
745         y=y2;
746     return (y);
747 }

748 /* Subrutinas SUCROS97 Goudriaan 1994 */
749 void gla(double Time,double Doyem,double DTeff,double Dvs,double Rgrl,double Deltat,double Sla,double
750 Lai,double Glv,double *CrecLai)
751 {
752     *CrecLai =Sla*Glv;
753     if ((Dvs < 0.3) && (Lai < 0.75))
754         *CrecLai=(Lai*(exp(Rgrl*DTeff*Deltat)-1.0))/Deltat;

755     if (Time <= Doyem)
756         *CrecLai=0.0;
757 }

758 void Assim(double Scp, double Amax,double Eff,double Kdf, double Lai, double Sinb, double Pardr, double
759 Pardf,double *Fgros)
760 {
761     double xGauss[5],wGauss[5];
762     double Sqv, Refh, Refs, Clustf, Kbl, Kdrt, Laic, Visdf, Vist, Visd, Visshd, Fgrsh, Vispp, Fgrsun, Vissun, Fgrs,
763     Fsla, Fgl;
764     int i,j;
765     xGauss[0]=0.0469101;
766     xGauss[1]=0.2307534;
767     xGauss[2]=0.5;
768     xGauss[3]=0.7692465;
769     xGauss[4]=0.9530899;
770     wGauss[0]=0.1184635;
771     wGauss[1]=0.2393144;
772     wGauss[2]=0.2844444;
773     wGauss[3]=0.2393144;

```

```

774   wGauss[4]=0.1184635;

775   Sqv=sqrt(1.0-Scp);
776   Refh=(1.0-Sqv)/(1.0+Sqv);
777   Refs=Refh*2.0/(1.0+2.0*Sinb);

778   Clustf=Kdf/(0.8*Sqv);
779   Kbl=(0.5/Sinb)*Clustf;
780   Kdrt=Kbl*Sqv;

781   *Fgros=0.0;
782   for (i=0;i<5;i++){
783       Laic=Lai*xGauss[i];
784       Visdf=(1.0-Refh)*Pardf*Kdf*exp(-Kdf*Laic);
785       Vist=(1.0-Refs)*Pardr*Kdrt*exp(-Kdrt*Laic);
786       Visd=(1.0-Scp)*Pardr*Kbl*exp(-Kbl*Laic);
787       Visshd=Visdf+Vist-Visd;
788       if (Amax > 0.0)
789           Fgrsh=Amax*(1.0-exp(-Visshd*Eff/Amax));
790       else
791           Fgrsh=0.0;
792       Vispp=(1.0-Scp)*Pardr/Sinb;
793       Fgrsun=0.0;
794       for (j=0;j<5;j++){
795           Vissun=Visshd+Vispp*xGauss[j];
796           if (Amax > 0.0)
797               Fgrs=Amax*(1.0-exp(-Vissun*Eff/Amax));
798           else
799               Fgrs=0.0;

800           Fgrsun=Fgrsun+Fgrs*wGauss[j];

801       }
802       Fslla=Clustf*exp(-Kbl*Laic);
803       Fgl=Fslla*Fgrsun+(1.0-Fslla)*Fgrsh;

804       *Fgros=(*Fgros)+Fgl*wGauss[i];
805   }
806   *Fgros=(*Fgros)*Lai;
807   }

808   void Astro(double Day, double Lat, double *Sc,double *Ds0, double *Sinld, double *Cosld, double *Dayl,
809   double *Dsinb, double *Dsinbe)

810   {
811   double PI=3.141592654;
812   double Radianes=PI/180.0;
813   double Aob, Aux2,Aux3;
814   double Dec;

815   Dec=-asin(sin(23.45*Radianes)* cos(2.0*PI*(Day+10.0)/365.0));
816   *Sinld= sin(Lat*Radianes)*sin(Dec);
817   *Cosld=cos(Lat*Radianes)* cos(Dec);
818   Aob=(*Sinld)/(*Cosld);
819   Aux2=(*Sinld)*(*Sinld);
820   Aux3=(*Cosld)*(*Cosld);
821   *Dayl = 12.0 * (1.0+2.0 * asin(Aob)/PI);
822   *Dsinb=3600.0*((*Dayl)*(*Sinld)+24.0*( *Cosld)*sqrt(1.0-Aob*Aob)/PI);
823   *Dsinbe=3600.0*((*Dayl)*(*Sinld)+0.4*(Aux2+0.5*Aux3)) + 12.0*( *Cosld)*(2.0+3.0*0.4*( *Sinld))*sqrt(1.0-
824   Aob*Aob)/PI);
825   *Sc = 1370.0 * (1.+0.033*cos(2.0*PI*Day/365.0));
826   *Ds0=(*Sc) * (*Dsinb);

```

```

827     }

828 void Totass(double Day, double Lat, double Dtr, double Scp, double Amax, double Eff, double Kdf, double
829 Lai, double *Dayl,double *Dtga, double *Ds0)
830 {
831     double xGauss[3], wGauss[3];
832     double Sc,Sinld,Cosld,Dsinb,Dsinbe,Sinb,Pard,Pardf;
833     double Par, PI;
834     double Hora, Atmtr,Frdf,Pardr, Fgros, Dtgas, Ds01, Dayl1;
835     int i;
836     PI =3.141592654;

837     xGauss[0]=0.112702;
838     xGauss[1]=0.50;
839     xGauss[2]=0.887298;
840     wGauss[0]=0.277778;
841     wGauss[1]=0.444444;
842     wGauss[2]=0.277778;

843     Astro(Day,Lat,&Sc,&Ds01,&Sinld,&Cosld,&Dayl1,&Dsinb,&Dsinbe);
844     Dtgas=0.0;

845     for (i=0;i<3;i++){

846         Hora=12.0+0.5*Dayl1*xGauss[i];
847         Sinb=max(0.0,Sinld+Cosld*cos(2.0*PI*(Hora+12.0)/24.0));
848         Par=0.5*Dtr*Sinb*(1.0+0.4*Sinb)/Dsinbe;
849         Atmtr=Par/(0.5*Sc*Sinb);
850         if (Atmtr <= 0.22)
851             Frdf=1.0;
852         else if (Atmtr > 0.22 && (Atmtr <= 0.35) )
853             Frdf=1.0-6.4*pow((Atmtr-0.22),2.0);
854         else
855             Frdf=1.47-1.66*Atmtr;
856         Frdf=max(Frdf,0.15+0.85*(1.0-exp(-0.1/Sinb)));
857         Pardf=Par*Frdf;
858         Pardr=Par-Pardf;
859         Assim(Scp,Amax,Eff,Kdf,Lai,Sinb,Pardr,Pardf,&Fgros);
860         Dtgas=Dtgas+Fgros*wGauss[i];
861     }
862     (*Dtga)=Dtgas*Dayl1*3600.0;
863     (*Dayl)=Dayl1;
864     (*Ds0)=Ds01;
865 }

866 void SubGrt(double Zrt,double Zrtm,double Dvs,double Tkl1,double Tkl2, double Tkl3, double Tkl4, double
867 Wcl1,double Wcl2,double Wcl3,double Wcl4, double Wcwp1,double Wcwp2,double Wcwp3,double
868 Wcwp4,double *Wsert)
869 {
870     *Wsert=1.0;
871     if ((Zrt < Tkl1) && (Wcl1 < Wcwp1))
872         *Wsert=0.0;
873     if ((Zrt > Tkl1) && (Zrt < (Tkl1+Tkl2)) && (Wcl2 < Wcwp2))
874         *Wsert=0.0;
875     if ((Zrt >(Tkl1+Wcwp3)) && (Zrt < (Tkl1+Tkl2+Tkl3)) && (Wcl3 <Wcwp3))
876         *Wsert=0.0;
877     if ((Zrt > (Tkl1+Tkl2+Tkl3)) && (Zrt < (Tkl1+Tkl2+Tkl3+Tkl4)) && (Wcl4 < Wcwp4))
878         *Wsert=0.0;
879     if (Dvs >= 1.0)
880         *Wsert=0.0;
881     if (Zrt > Zrtm)
882         *Wsert=0.0;

```

```

883     }

884     void SubFr(double Wcl,double Wcfc,double P, double Wcwp, double Wcwet,double Wcst, double *Wse)

885     {
886     double Fr,Wccr;

887     Wccr=Wcwp+(1.0-P)*(Wcfc-Wcwp);
888     if (Wcl > Wcwet)
889         Fr=(Wcst-Wcl)/(Wcst-Wcwet);
890     else if ((Wcl <= Wcwet) && (Wcl > Wccr))
891         Fr=1.0;
892     else if ((Wcl <=Wccr) && (Wcl > Wcwp))
893         Fr=(Wcl-Wcwp)/(Wccr-Wcwp);
894     else
895         Fr=0.0;
896     *Wse=min(1.0,max(0.0,Fr));
897     }

898     double limit(double ylow, double yhigh,double x)
899     {
900     double y;
901     if ((x >= ylow) && (x <=yhigh))
902         y = x;
903     else if (x < ylow)
904         y=ylow;
905     else if (x > yhigh)
906         y=yhigh;
907     return (y);
908     }

909     double max1(double a,double b, double c, double d)
910     {
911     double Maximo;
912     Maximo=max(a,b);
913     Maximo=max(Maximo,c);
914     Maximo=max(Maximo,d);
915     return (Maximo);
916     }

917     double min1(double a, double b,double c)
918     {
919     double Minimo;
920     Minimo=min(a,b);
921     Minimo=min(Minimo,c);
922     return (Minimo);
923     }
924     double NotNull(double x)
925     {
926     double y;
927     if (x==0.0)
928         y=1.0;
929     else
930         y=x;
931     return (y);
932     }

933     /*=====
934     * See sfuntmpl.doc for the optional S-function methods *
935     *=====*/

936     /*=====

```

```
937  * Required S-function trailer *
938  *=====*/

939  #ifdef MATLAB_MEX_FILE /* Is this file being compiled as a MEX-file? */
940  #include "simulink.c" /* MEX-file interface mechanism */
941  #else
942  #include "cg_sfun.h" /* Code generation registration function */
943  #endif
```