Determinación del costo de operación para una flota de vehículos eléctricos

Pablo Chiesa, Matías Iglesias, Álvaro Castro
Instituto de Ingeniería Eléctrica
Facultad de Ingeniería de la Universidad de la República Oriental del Uruguay

IMPORTANTE: Este trabajo se realizó en el marco del curso Simulación de Sistemas de Energía Eléctrica - SimSEE y fue evaluado por el enfoque metodológico, la pericia en la utilización de las herramientas adquiridas en el curso para la resolución del estudio y por la claridad de exposición de los resultados obtenidos. Se quiere dejar expresamente claro que no es relevante a los efectos del curso la veracidad de las hipótesis asumidas por los estudiantes y consecuentemente la exactitud o aplicabilidad de los resultados.

Contenido

1. Introducción

La electrificación del sistema de transporte facilitará la integración de las energías renovables en el sistema eléctrico ya que resuelve muchos de los problemas que se plantean en la actualidad y que algunos se agravarán en el futuro.

La introducción de autos eléctricos permitirá a largo plazo:

- Gestionar las energías eólicas y solares.
- Almacenar la energía eléctrica.
- Tratar los excedentes de energía. Es el caso de la energía eólica, cuando la generación supera la demanda, generalmente en horas valle. Para reducir estos excedentes, se podrían adoptar algunas de las siguientes medidas:

1. Bombeo: no es una tarea sencilla y presenta numerosos problemas ambientales.
2. Almacenamiento de la energía sobrante: a través de grandes acumuladores o supercondensadores (tecnología en desarrollo).
3. Cargar baterías de vehículos eléctricos.

Por lo expuesto anteriormente es preciso analizar qué sucederá en nuestro país con la introducción de los autos eléctricos en la demanda. De esta manera también se abre una posibilidad para UTE de poder ingresar en un nuevo mercado, el transporte.

Existen países como Noruega (véase Figura 1) en donde está tecnología se aplica desde 2004 y va en aumento exponencial, lo que demuestra la aceptación por parte del mercado, además sirve para solucionar los problemas antes mencionados. Es importante para nosotros ver cómo evolucionan estas tecnologías en los mercados del primer mundo y una manera de tomar ejemplo de países y sistemas mas desarrollados que el nuestro.

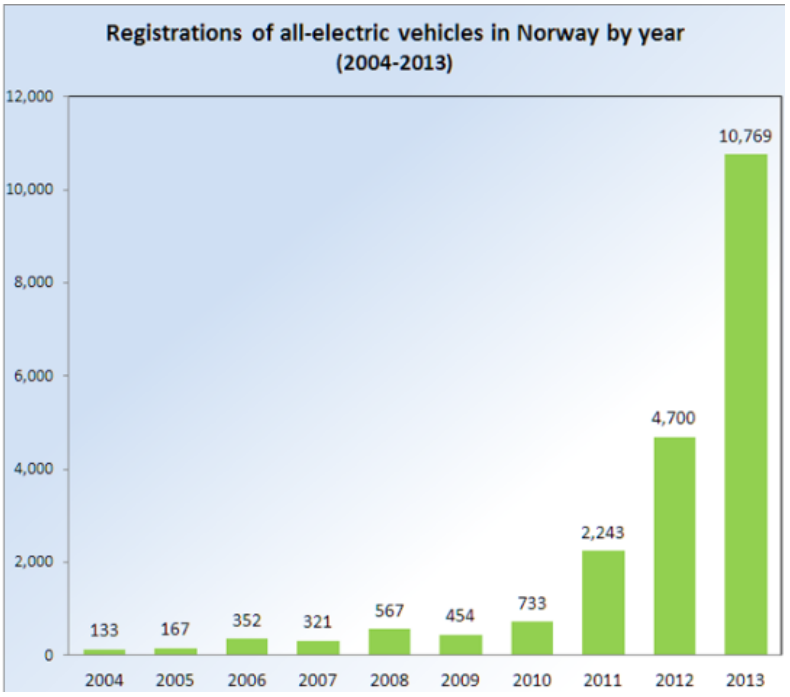


Figura 1: Autos eléctricos de Noruega en cifras 2004-2013

Para el presente estudio se utilizaron modelos de autos como el Nissan Leaf y el Citroen-c-zero los cuáles se pueden observar en las figuras 2 y 3, respectivamente:

Figura 2: Nissan Leaf

Figura 3: Citroen-c-zero

2. Objetivo

El presente trabajo tiene como objetivo inicial estudiar la demanda anual del 2014 de Uruguay y ver los efectos que generaría los autos eléctricos en dicha demanda, y determinar los costos. Para eso se asumirá un modelo eléctrico para el "auto eléctrico", de aquí en más **AE**, y se realizarán corridas utilizando el software provisto en el curso.

Estudiar cómo se comportan las curvas de carga para los vehículos eléctricos y cómo afecta este comportamiento al costo total de operación del sistema.

Dentro de SIMSEE se crean las fuentes correspondientes para anexarlas a la demanda actual teniendo en cuenta las diferentes tecnologías, y dos diferentes clases. La primera de ellas utilitaria que corresponde a los vehículos que recorren más kilometraje diario y por ende necesitan más recargas de energía. Por otro lado, la segunda clase, denominada "familiar", se compone de vehículos de menor kilometraje diario.

La solución considerada permite flexibilidad para modificaciones futuras. Se realizan corridas de corto y largo plazo según la cantidad de vehículos de cada clase correspondiente a simular en el sistema.

Se crea una herramienta para relevar las diferentes tecnologías (basado en el mercado actual). Los datos de salida se acondicionan en una plantilla SimRes3 conveniente para nuestro propósito, donde se imprime el despacho total y se indica el costo, esto nos permite comparar contra la situación actual (sin vehículos eléctricos) y estudiar los diferentes incrementos de costo de operación según los distintos casos de estudio.

3. Tecnologías de vehículos eléctricos utilizados

Las hipótesis de trabajo que fueron realizadas acerca del vehículo se describen a continuación:

1. Los vehículos analizados son de tipo eléctrico cuya conexión se puede realizar mediante un enchufe.

2. Las baterías de los vehículos son modeladas teniendo en cuenta los principales vehículos eléctricos que existen en el mercado. Se optó por tomar un valor representativo dentro de este rango, como se puede observar en la siguiente figura:

Modelo	Autonomía (kWh)	Autonomía (km)	$kWh_{Batería}/100km$
Reva L-ion[4]	11	120	9,17
Think City[5]	25	200	12,50
Mitsubishi i-Miev[6]	16	130	12,31
Citröen C-Zero[7] [8]	16	130	12,31
Renault Fluence ZE[9]	22	160	13,75
Nissan Leaf[10]	24	160	15,00
Tesla Roadster 42	42	257	16,34
Tesla Roadster 70[11]	70	483	14,49
MEDIA	28,25	205	13,78

Figura 4: Modelos de Mercado

3. Las baterías tienen diferentes capacidades de carga y se modelan dentro del sistema *calcular* para condicionar el proceso markoviano que describe el transcurrir de la suerte del vehículo durante el día.

4. Hipótesis de trabajo para la estimación de cargas

Los autos eléctricos se modelarán como cargas que demandan una potencia cuando están conectados a la red de distribución de energía eléctrica. El modelado de los mismos es a través de un proceso markoviano condicionado, el cuál se puede simplificar mediante el esquema de la figura 5:

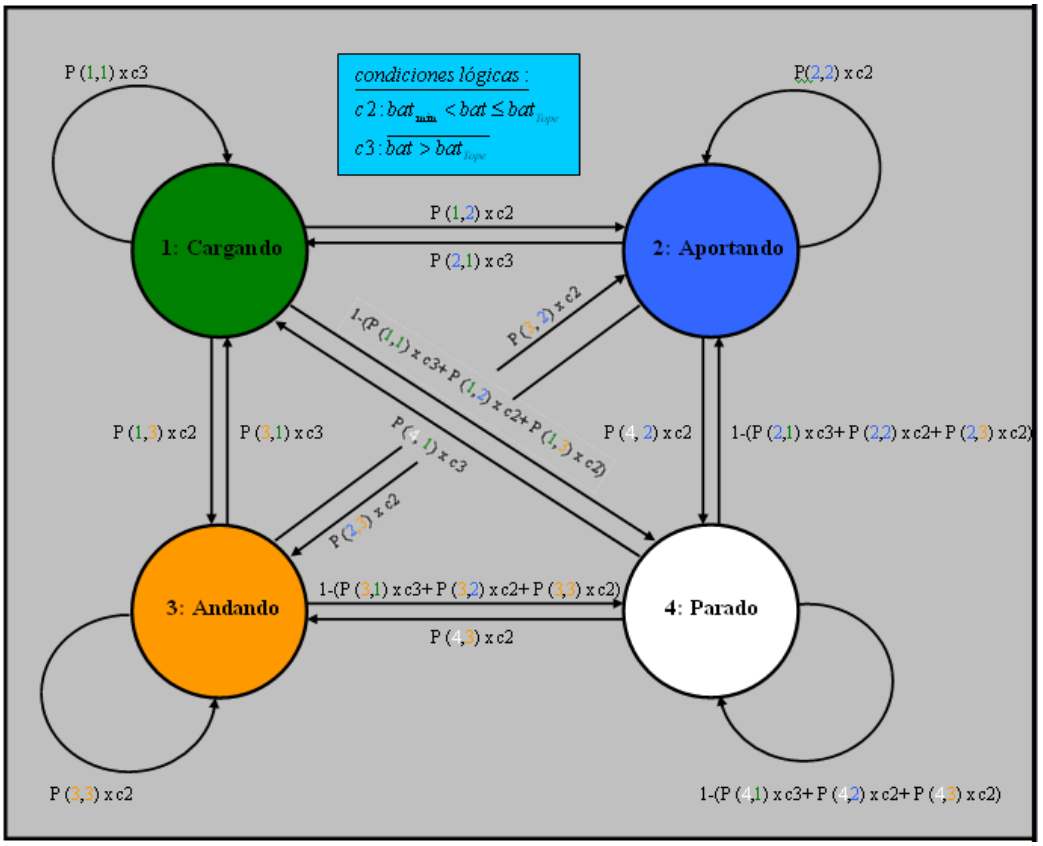


Figura 5: diagrama de Markov condicionado

Tal como se puede observar en la figura 5 el modelado se realiza en cuatro estados bien diferenciados, a saber:

- Estado 1: cargando
- Estado 2: aportando
- Estado 3: andando
- Estado 4: parado

Cada auto tiene asociada una matriz de transiciones del proceso de markov en la cuál se indica en el elemento de la fila i y columna j, la probabilidad de transición del estado i al estado j. Claramente la suma de los elementos por filas de la matriz es uno.

Cabe destacar que este proceso de markov de transición de estados esta condicionado por variables booleanas que habilitan o deshabilitan la transición a un determinado estado dependiendo de la condición de la batería del vehiculo.

Cada vehículo además pasa por un proceso markoviano que decide a que tipo de tecnología pertenece, es decir, para cada vehículo a simular se sortea primero a qué modelo pertenece. Lo anterior influye en la cantidad de KWh máxima almacenable en su batería y la cantidad de horas de carga. Para este proceso tenemos en cuenta datos de autos actualmente existentes en el mercado y considerados como los más vendidos en los últimos años (autos totalmente eléctricos).

Para ajustar aún más a la realidad los valores de ambas matrices, necesitaríamos conocer datos precisos de cuantos autos utilitarios y/o familiares están andando o están parados en determinadas horas del día y en determinados meses del año. A su vez, se debe conocer cómo son las ventas de cada tecnología, para saber cuántos autos de cada modelo circulan por las calles y así ajustar aún más el proceso markov que decide el modelo asignado a cada vehiculo simulado.

Adjunto a esta documentación se encuentran dos ejemplos grabados en video de cómo utilizar la interfaz de calcular, según una simulación de pocos días (para ver cómo quedan las cargas en el correr de las horas) y otra simulación con más días.

Lo que se intenta resolver es el consumo de una cantidad determinada de autos en una cantidad determinada de días, estudiar cómo son las variaciones dependiendo de las diferentes tecnologías (marcas, modelos) y dependiendo de la variación del tiempo de carga. Para realizar esto se divide cada día simulado en tres regiones horarias, un primer tramo del día comprendido entre las 0AM y las 6AM, otro desde las 6AM hasta las 6PM y un último tramo horario desde las 6PM hasta las 12PM. Entonces es necesario definir tres matrices de transición de estados de Markov, una por cada tramo horario. Los valores de estas tres matrices son fundamentales a la hora de decidir qué es lo que realizará el auto simulado en la siguiente hora simulada.

Para tener más control sobre los datos simulados, se considera que cada auto puede pertenecer a un número fijo de distintas tecnologías o modelos (en el caso de este trabajo se utilizaron cuatro modelos diferentes), por consiguiente en nuestro caso los vectores que definen las tecnologías `tecEE` (guarda la máxima capacidad acumulable en las baterías) y el vector tecDisferentes (guarda la cantidad de horas de carga)

```
tecEE=[24 24 25 16];  % KWh de batería

en este caso se consideran las marcas:

Nissan Leaf
Renault Fluence
Think City
Mitsubishi i-Miev

tecDisferentes=[6 8 6 7];   %horas de carga (pasos de carga de cada
marca distinta)
```

El lector se podría preguntar por qué hacer referencia a una tecnología y no simplemente a una marca o modelo. En principio se podría querer simular tanto autos como cualquier otra cosa eléctrica: motos, camiones, ómnibus, etc, eso se define justamente mediante dos parámetros, el primero es la capacidad máxima almacenable en las baterías y el segundo es el tiempo que demora en almacenar esa capacidad máxima. Se considera que durante el paso de una hora, la potencia de conexión a la red es el cociente entre la energía máxima almacenable y el paso de carga correspondiente, por ejemplo, para el modelo ***Nissan Leaf*** consideramos una potencia de conexión constante de 24/6 = 4kW , cada auto ***Nissan Leaf*** que aparezca en la población total consumirá 4 kWh de energía a la red durante el paso de esa hora de simulación si se encuentra en el estado 1) cargando , si se encuentra en el estado 2) entregando, se considera un aporte de -4kWh de energía a la red.

Podemos ver una corrida del *calcular* en donde nos muestra las curvas para pocos días, como varían las cargas dependiendo del tramo horario. Nuestro simulador de cargas *calcular* , tiene en cuenta que el día se divide en 3 grandes franjas horarias, pero cabe destacar que esto no es tan así. Esto varia muchísimo según la estación del año en la que estemos y además deberíamos considerar otros coeficientes para casos de días feriados y fin de semana. Como veremos el caso de fin de semana esta incluido definiendo otra fuente SimSee que lo contemple , mediante una fuente periódica que incrementa o atenúa los valores de demanda globales obtenidos por la flota, pero para hacerlo mas preciso seria necesario incluir estas consideraciones desde los coeficientes de Markov mismo del proceso por cada auto simulado.

Otro comentario que queremos establecer , es que por cada modelo o tecnología diferente que estamos simulando, se necesitaría una potencia de conexión distinta que se adapte a esa tecnología, lo que haría complicado para la distribuidora tener muchos puntos de conexión diferentes y para los usuarios que quieran cambiar su modelo de auto, puesto que deberían cambiar también el punto de conexión.

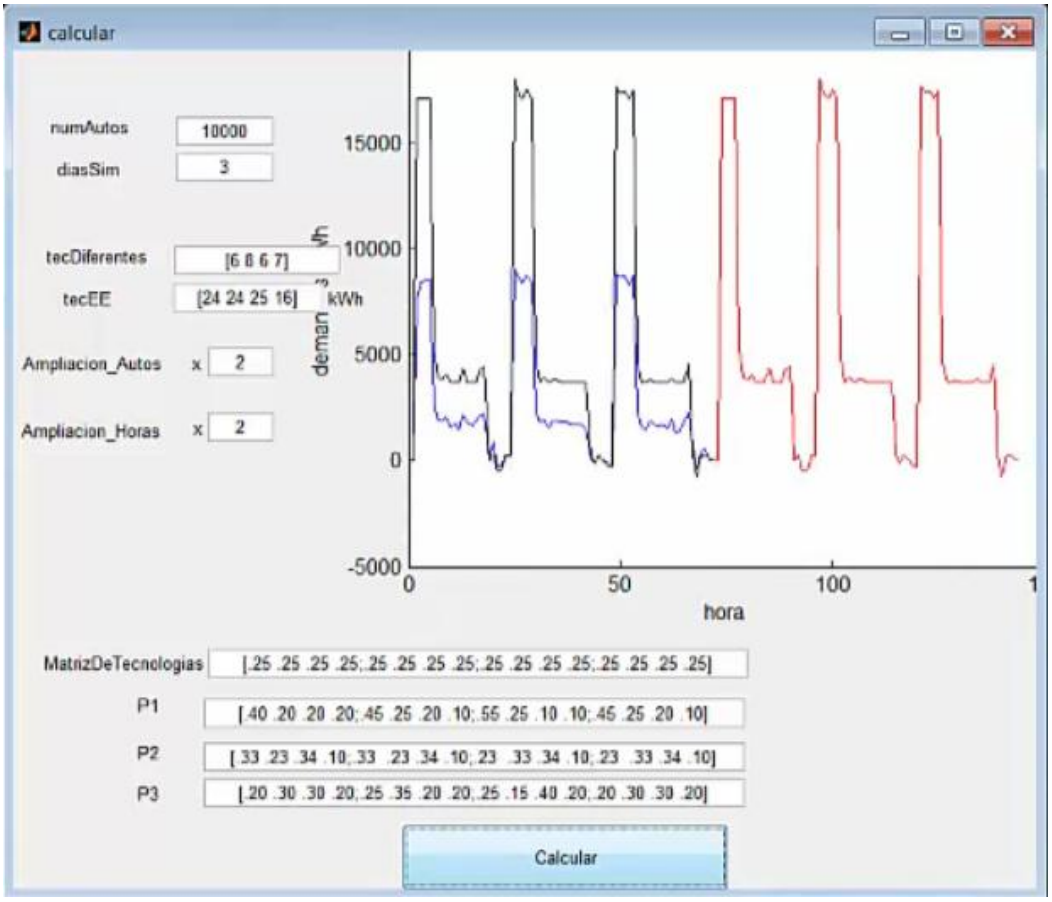


Figura 11: simulación con *calcular*

En la figura 11 se puede ver que las curvas azules son las simuladas, las curvas en negro son las expansiones utilizando el algoritmo de **K-MEANS**, para expandir por 2 en este caso el numero de autos, y las rojas son las expansiones temporales, en donde simplemente se copia la serie expandida, en este caso también por 2 en la cantidad de horas. Este sistema permite simular una gran cantidad de autos en una gran cantidad de días. También podemos ver cómo se reproducen estas expansiones utilizando **K-MEANS** en las gráficas monótonas de demanda (simplemente un “sort” u ordenamiento de menor a mayor en el valor de las demandas), ver grafica siguiente:

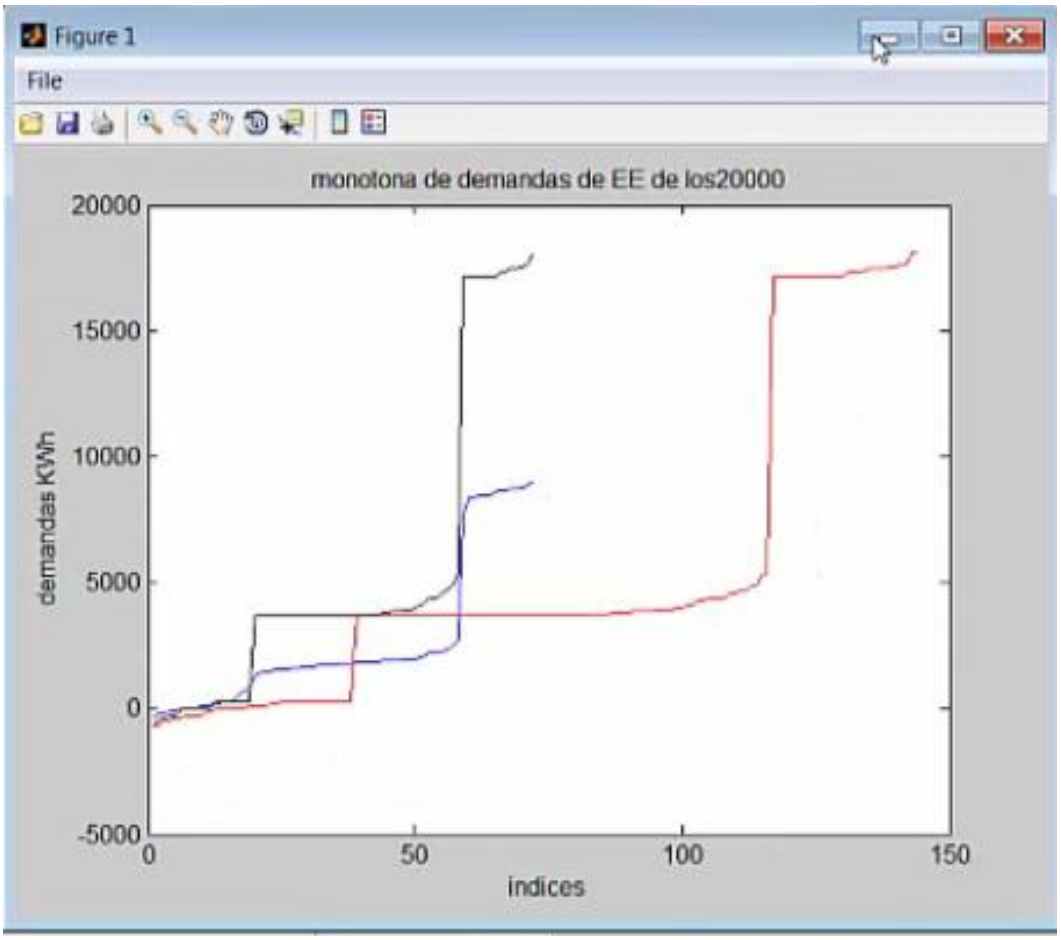


Figura 12: monótonas de demanda tras realizar las expansiones

5. Hipótesis para el armado de la sala en SimSee

Se crea una fuente Combinación con dos clases de vehículos eléctricos, una clase correspondiente al vehículo “familiar” de poco kilometraje diario y otra clase representada por el vehículo “utilitario o de servicios” como ser las camionetas de reparto, los taxis, remises, entre otros.

Para crear las series y las fuentes CEGH de estas dos clases, se utilizan los datos del siguiente cuadro extraído del Uruguay en cifras 2013 - Instituto Nacional de Estadística:

TRANSPORTE Y COMUNICACIONES

Parque automotor del país según tipo de vehículo utilizado.

	2010	2011
Total	**1.471.875**	**1.604.464**
Autos y camionetas	612.368	649.502
Camiones y tractores	64.640	67.982
Remolques y semi-remolques	34.057	27.891
Ómnibus y minibuses	8.218	8.639
Taxis y remises	6.177	6.281
Motos y ciclomotores	746.415	844.169

Fuente: Ministerio de Transporte y Obras Públicas (MTOP) - Dirección Nacional de Transporte, con datos de las Intendencias.
Nota: No se cuenta con la información de todas las Intendencias. El valor indicado se calculó tomando la última información disponible. Por ello el valor indicado no se corresponde exactamente con el parque vehicular total registrado en Intendencias.

Figura 11: Cifras del INE

Se considera una población de vehículos eléctricos del 10% de la totalidad según el tipo de vehículo, por lo que serían aproximadamente 64950 autos y camionetas.

Se estima un uso de 40 kilómetros diarios por auto para la clase “familiar” luego se considera aproximadamente 200 kilómetros diarios para la clase “utilitarios”, estos son valores aproximados, pues el kilometraje es aleatorio y con una gran dispersión viéndolo en su totalidad.

Al crear los modelos de demanda se debe especificar las matrices de Markov para los procesos por separado, una cosa es para el promedio de autos familiares a un cierto horario y otra distinta son las probabilidades de cambio de estado en ese horario para un taxi por ejemplo.

Supongamos que el taxi realice 250 kilómetros en dos turnos diarios , mientras que un auto familiar realizando trayectos cortos y rutinarios dentro de la semana quizás recorra a lo sumo 40 o 50 kilómetros. Estos son los datos de entrada al sistema generador de demandas para los vehículos a través de la definición del tipo de Tecnología (pasos de carga, máxima energía almacenable en batería) y de las matrices de Markov para modelar el comportamiento hora a hora de cada uno de los vehículos.

Se arman dos fuentes, una para clase familiar de 65kautos (10% del total actual según estadísticas de transporte) y unos 25Kautos utilitarios, que comprenden a los 6000 taxis y remises. Además, se crean las series CEGH, correspondiente a los autos familiares y los utilitarios. Posteriormente se arma una fuente combinación que se puede expresar de la siguiente forma:

$$C_1 \times 65000 \times AE_f + C_2 \times 25000 \times AE_u$$

- AE_f corresponde a la cantidad de autos eléctricos familiares.
- AE_u corresponde a la cantidad de autos eléctricos utilitarios.

Los coeficientes C_1 y C_2 son los que determinan los cambios que se producen en la demanda de los autos eléctricos y por ende influyen sobre el costo. Inicialmente se considera que valen: $C_1 = 0{,}75$ y $C_2 = 0{,}8$.

A continuación se presenta solo un ejemplo para 150KAutos (caso de mayor cantidad simulada) :

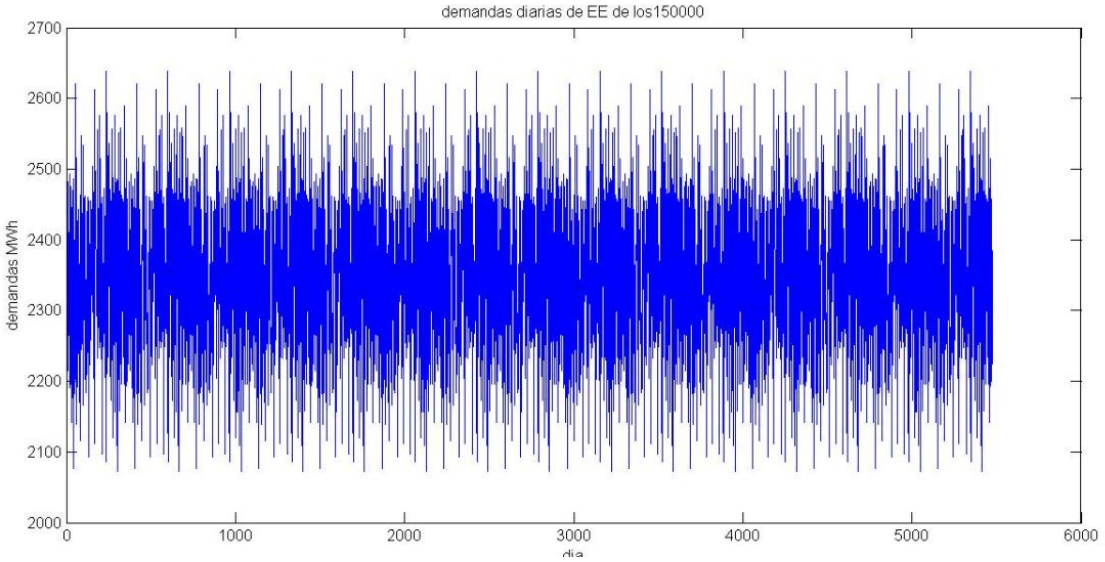


Figura 12: Demanda de los 150KAE

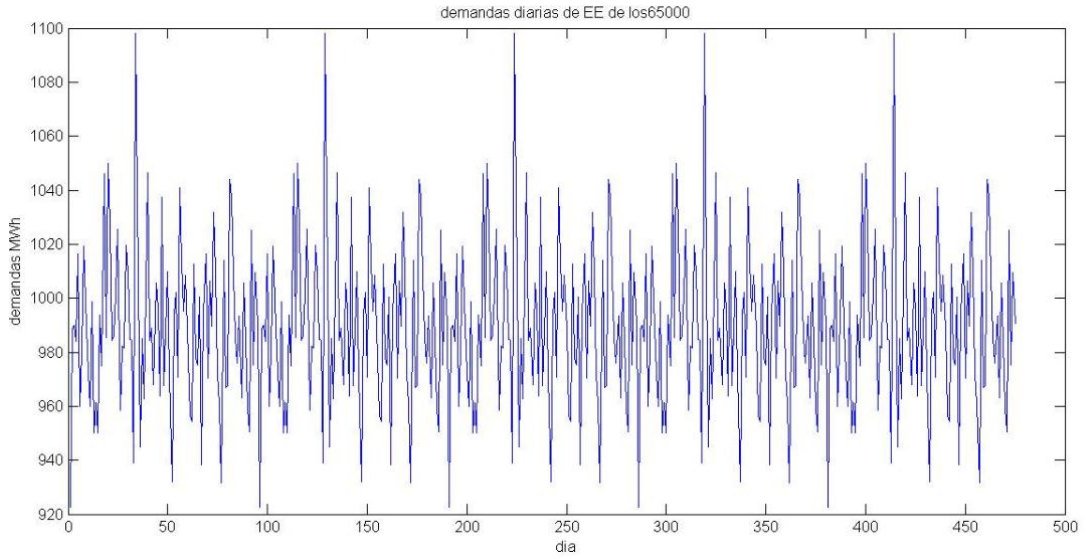


Figura 13: Demanda de los 65KAE familiar

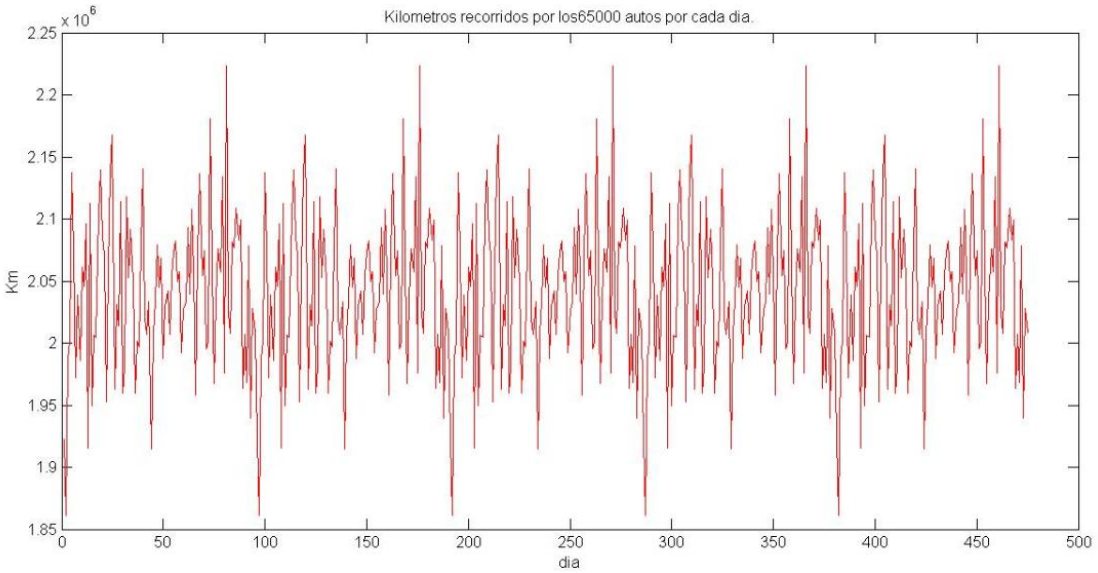


Figura 14: Demanda de los 65KAE familiar

Como vemos, es importante tener el dato de la autonomía por modelo, y así conocer cuantos kilómetros recorre la flota, de esta manera sabemos si los coeficientes de Markov se ajustan a la realidad (insistimos que aun así el uso de un vehículo es muy variado caso a caso) pero en términos generales podemos estimar un kilometraje diario por clase familiar/utilitaria.

Una vez obtenida la serie de autos se arma el archivo de series editándolo adecuadamente a cada caso.

```
1       NSeries
2014    1       1       0       0       0  //
1   // Período de muestreo en horas
2000    NPuntos
250     Puntos por ciclo
        Demanda_25kAutos_Utilitarios

1       154368
2       207361
3       178612
4       185639
5       183585
6       171004
7       189533
8       154946
9       180543
10      190232
11      180598
12      205567
13      159232
14      181142
15      170506
16      175029
17      189459
18      193653
19      182118
20      199393
21      207931
22      194542
23      165737
24      164015
25      185179
```

Estos datos de demanda por día se expresan en kWh.

Se carga en el análisis serial y se obtienen los archivos CEGH para cada clase de auto. A continuación se agrega una fuente Sintetizador CEGH y se multiplica por una fuente constante de valor 1/1000, para pasar a MWh de energía. Posteriormente, se crea otra fuente periódica constante de valor 0,75 para modelar el comportamiento de la demanda de autos eléctricos en los fines de semana. Esta fuente de fin de semana tiene largo 7 días, en donde estará desactivada en 5 días y en los restantes 2 estará activa,

Se compara la situación actual, llamada caso BASE y la situación en la que si existen autos eléctricos que se ha estudiado en cinco escenarios distintos:

- 65 kAutos familiares y ningún auto utilitario.
- 65 kAutos familiares y 25 kAutos utilitarios.
- 25 KAutos utilitarios y ningún auto familiar.
- 12,5 kAutos utilitarios y 32,5 kAutos familiares.
- 12,5 kAutos utilitarios y 32,5 kAutos familiares. Simulados 5 años

Otras hipótesis sobre la SALA son:

- Se tomó un solo nodo (UY).
- Demanda base: A partir de 2013 , demandaBase2013.bin
- Eólica: Utilizamos el archivo de fuentes(CEGH_Eol1MW_6puntos.txt) y consideramos las siguientes unidades disponibles a lo largo del tiempo:

Fecha de Inicio	Número de máquinas	Periodica?	Capa			
Auto	1200	NO	0			
01/05/2015	1200	NO	0			
01/11/2015	1500	NO	0			
01/01/2020	2200	NO	0			

Figura 15: Unidades disponibles para Eólica

- Solar Fotovoltaica: se utiliza una fuente llamada KT_Solar (CEGH_Kt_horaria_8puntos.txt) con 8 bornes. Aquí se considera que existe una planta PV en cada uno de los ochos lugares del país correspondientes a los ocho bornes, totalizando: 200MW.

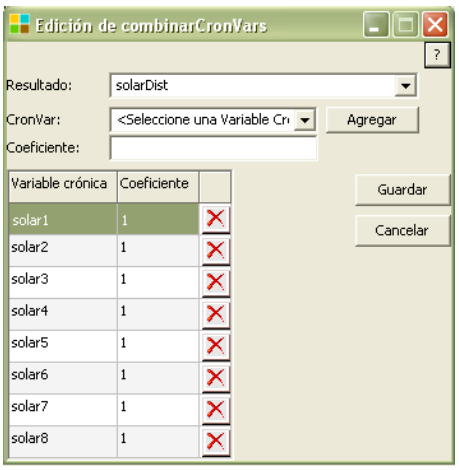


Figura 16: Solar Distribuida

6. Resultados Obtenidos en el SimSee

Caso Base:demanda_base2013+Aratiri

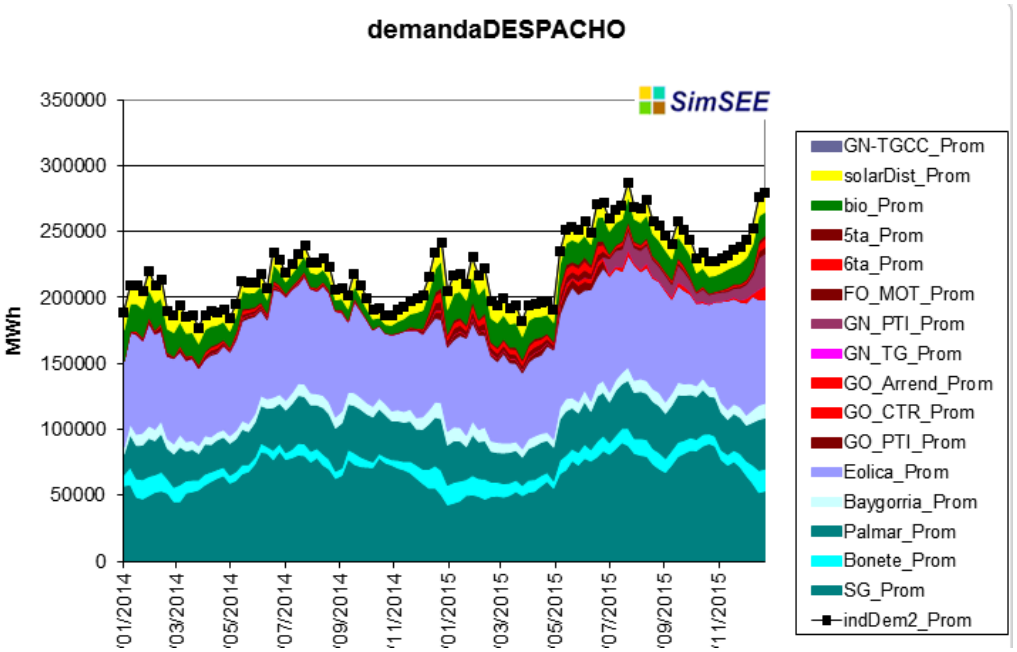


Figura 17: Demanda Caso Base

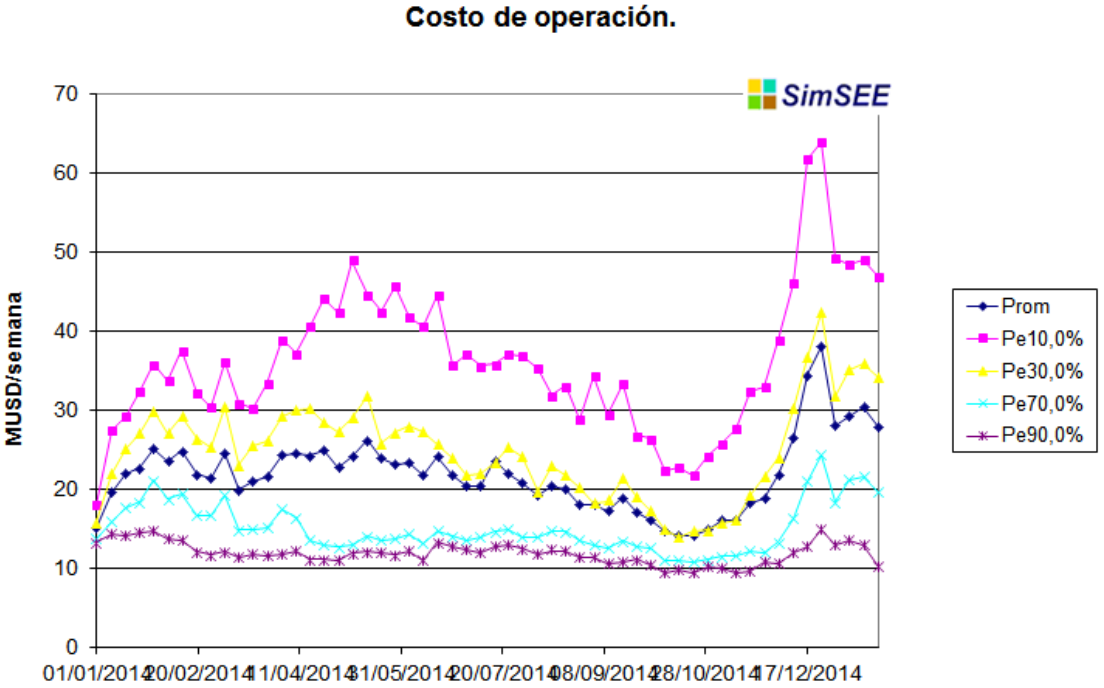
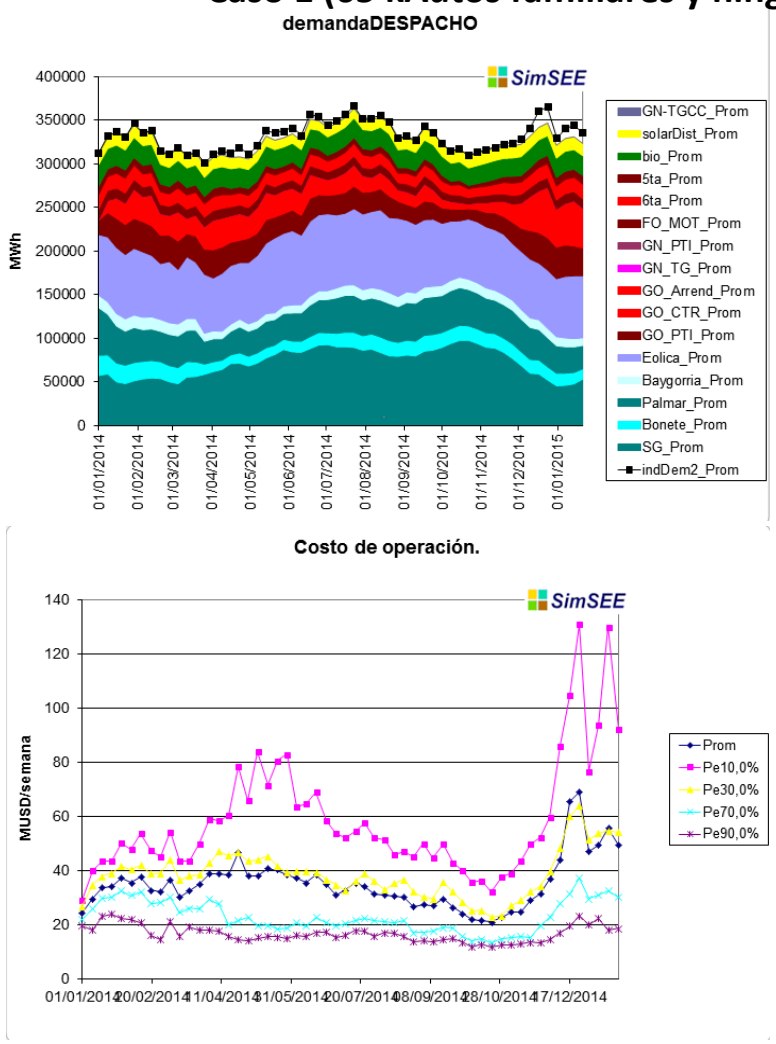


Figura 18: Costo de Operación

Casos de estudio:

- **Caso 1 (65 kAutos familiares y ningún auto utilitario)**



- $C_1 \times 65000 \times AE_f + C_2 \times 25000 \times AE_u$

  C_1 =1

  C_2 =0

Figura 19: Despacho y Costos marginales de Operación Caso 1

- **Caso 2 (65 kAutos familiares y 25 kAutos utilitarios)**

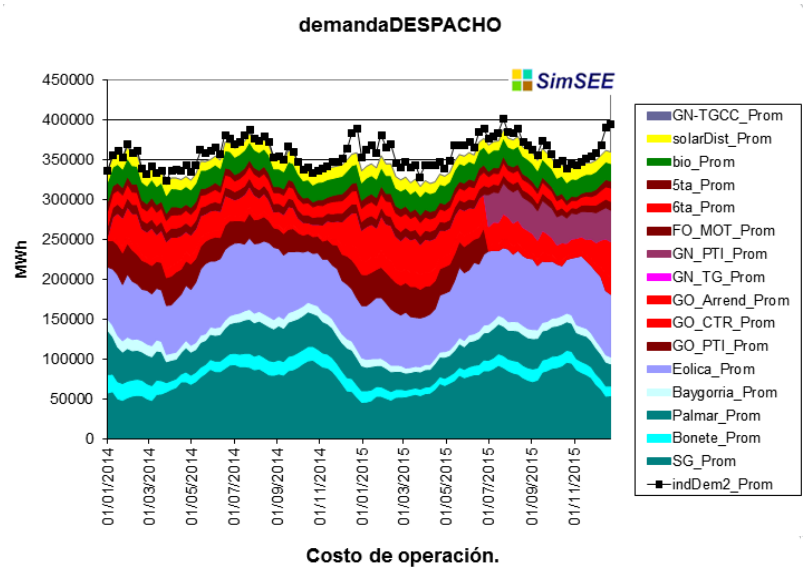


Figura 20: Despacho y Costos marginales de Operación Caso 2

- $C_1 \times 65000 \times AE_f + C_2 \times 25000 \times AE_u$

  C_1=1

  C_2=1

- **Caso 3 (25 kAutos utilitarios y ningún auto familiar)**

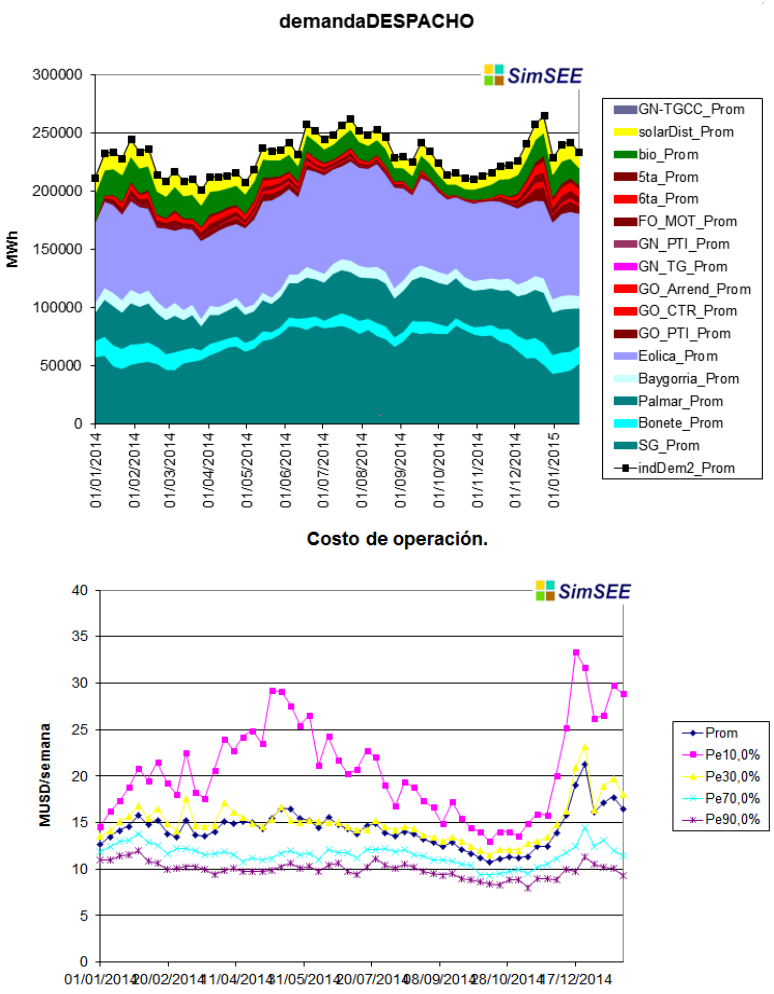


Figura 21: Despacho y Costos marginales de Operación Caso 3

- $C_1 \times 65000 \times AE_f + C_2 \times 25000 \times AE_u$

  C_1=0

  C_2=1

- **Caso 4 (12,5 kAutos utilitarios y 32,5 kAutos familiares)**

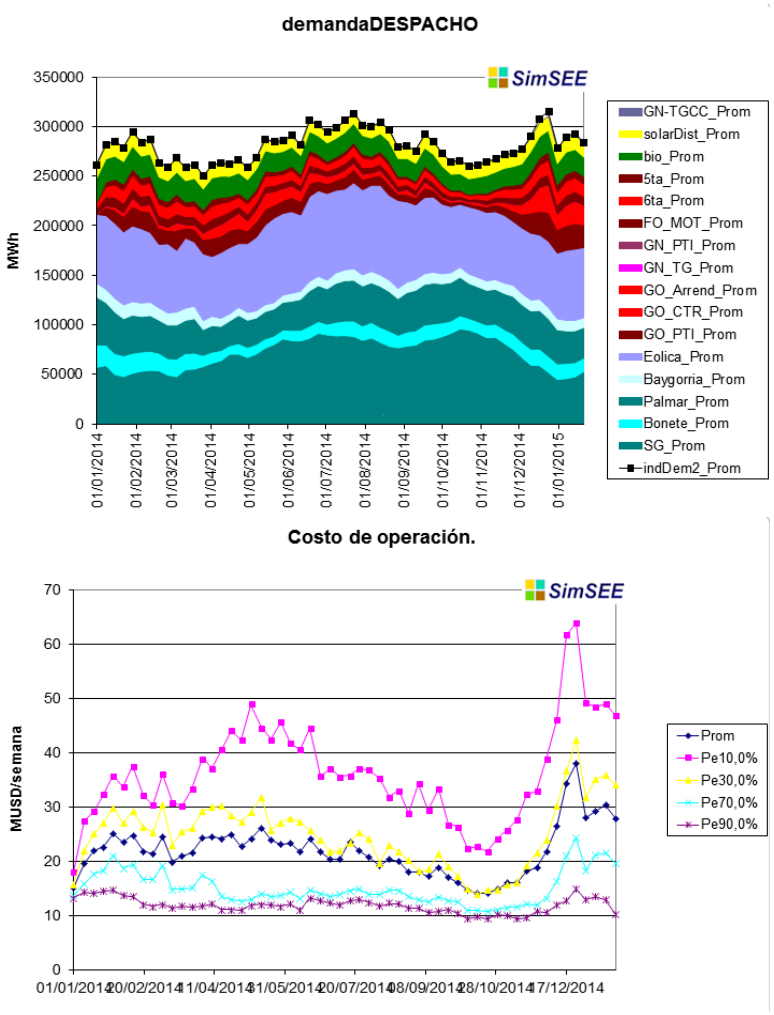


Figura 22: Despacho y Costos marginales de Operación Caso 4

- $C_1 \times 65000 \times AE_f + C_2 \times 25000 \times AE_u$

  C_1=0.5

  C_2=0.5

- **Caso 5 (12,5 kAutos utilitarios y 32,5 kAutos familiares) simulación a 5 años**

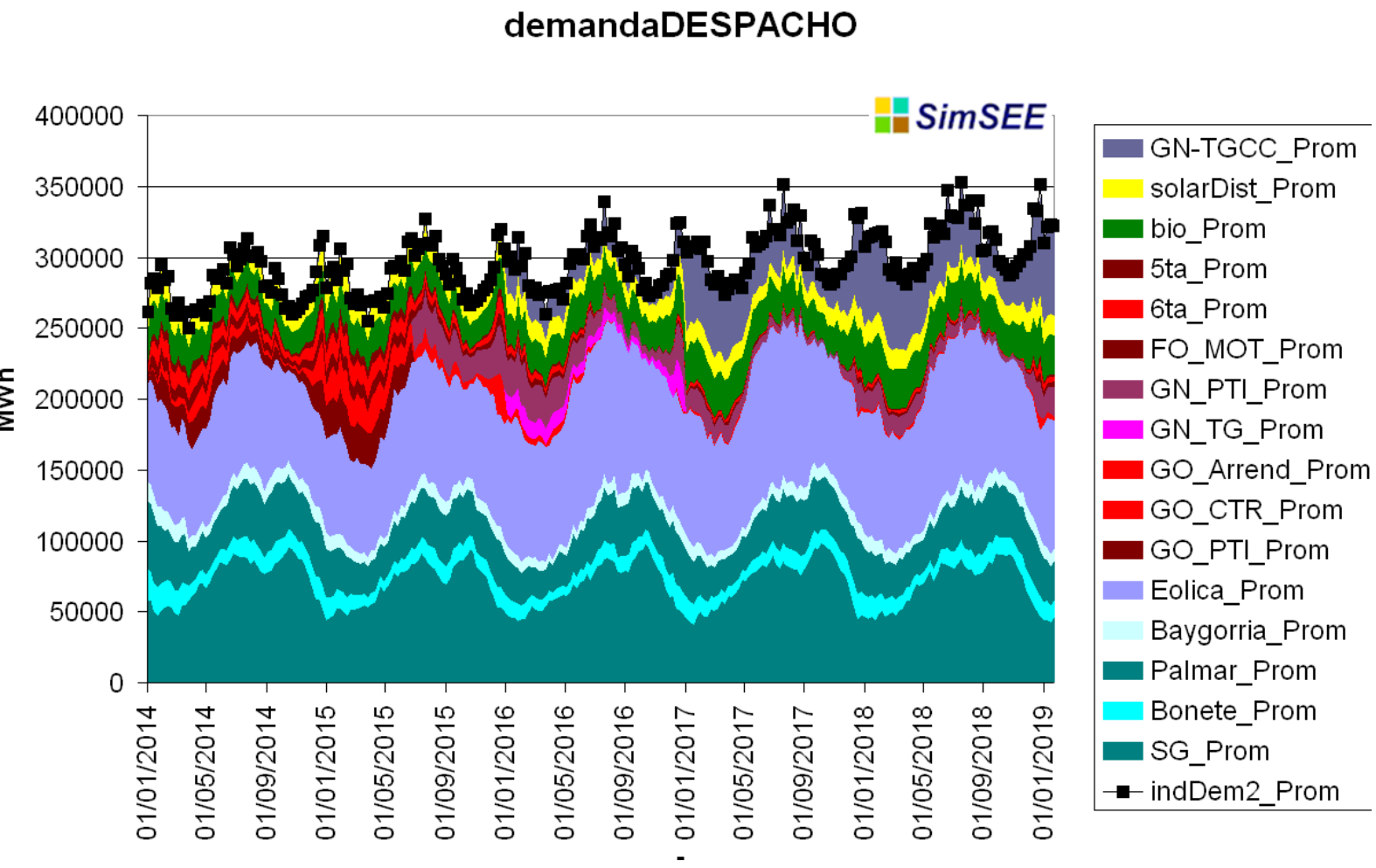


Figura 23: Despacho simulación a 5 años con paso semanal

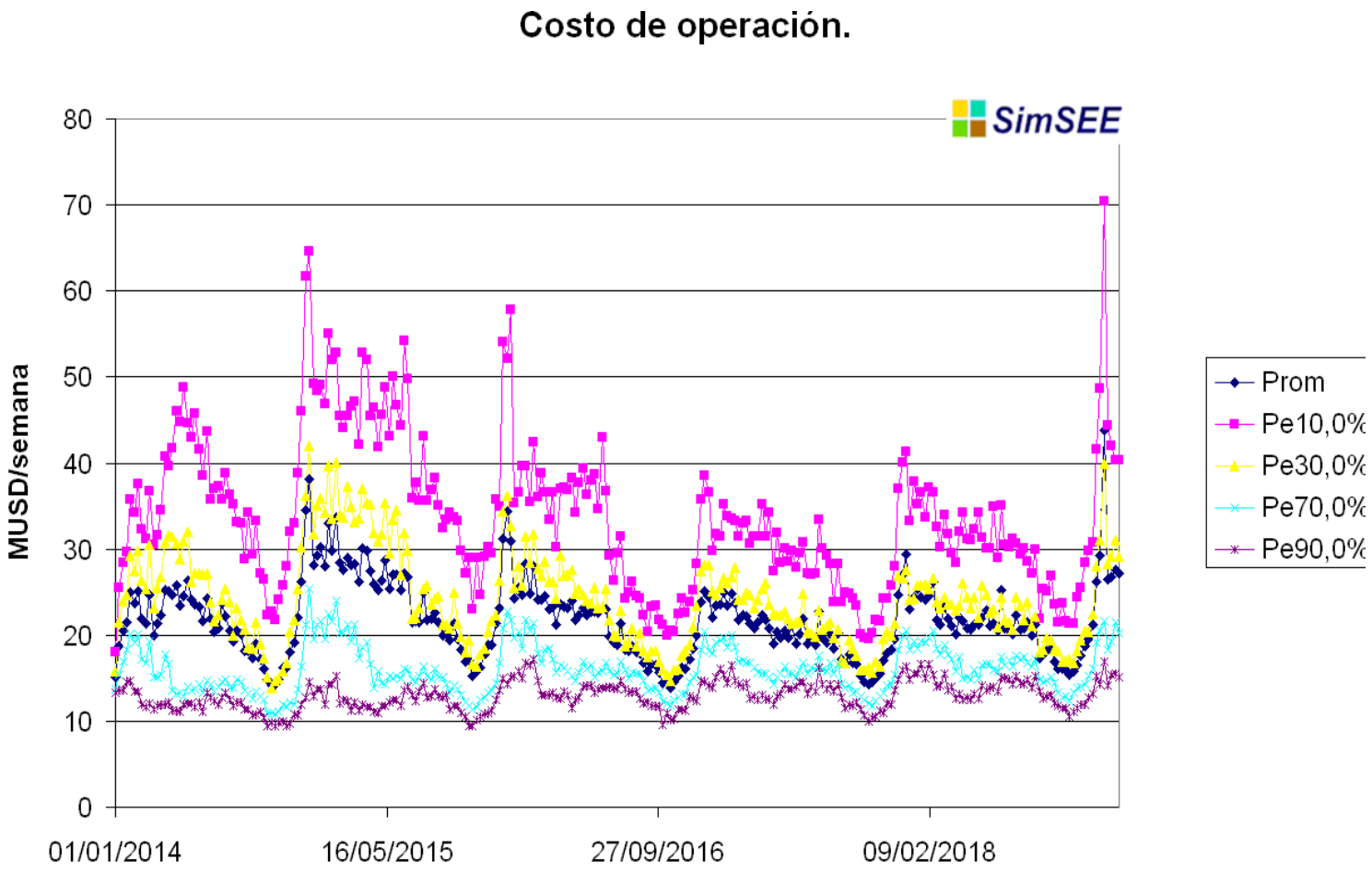


Figura 24: Costo de Operación de la simulación a 5 años con paso semanal

CONCLUSIONES

Como vemos en todos los casos observados, aun en el más exigente (caso 2), desde el punto de vista de la demanda de energía, el sistema con estas hipótesis realizadas logra dar el despacho sin recurrir a falla.

Pudimos familiarizarnos con la creación de fuentes CEGH, utilizar fuentes producto, fuentes combinación, fuentes periódicas. Crear los archivos de serie necesarios, utilizando los conceptos de mini-Ciclos adecuados a nuestros casos según los archivos de salida de demandas producidos por nuestra aplicación *calcular.* En dicha aplicación pudimos estudiar el caso de demandas variables según tramos horarios, estudiar cómo se condicionan las transiciones de estados de proceso markoviano y también utilizar algoritmos de "medias locales" como el K-Means, que nos permitieron obtener series de datos expandidas de manera rápida y con perdida minima de información.

Creemos que es un inicio para modelar cualquier tipo de carga que pueda ayudarnos a manejar los excedentes de energía entregados por las energías renovables, principalmente la eólica. Lo mismo se aplicaría por ejemplo a concentradores de calor, estudiando más en detalle su funcionamiento se podría adaptar los pasos de carga y la cantidad de energía acumulable a este tipo de dispositivos.

ASPECTOS A MEJORAR

Tener más certeza a la hora de modelar las curvas de carga de cada una de las baterías de los modelos involucrados ya que estas no son lineales. Tener también más certeza sobre los coeficientes de Markov ya comentados, para las diferentes épocas del año o días feriados por ejemplo.

Incluir los vehículos híbridos y modificar las características del modelado como para contemplarlos, pensar un esquema de gestión de cargas con tarifas inteligentes, adaptarlo al enfoque SmartGrid para lograr aplanar las curvas de demanda durante el día y utilizar los autos para este propósito (gestión de demandas).

Poder modelar mejor las incorporaciones en el paso del tiempo de los diferentes actores, principalmente el actor solar.

Poder estudiar el concepto de micro generación y de NegaWatts y adaptarlo al uso de carga para los autos eléctricos mediante un control inteligente.

ANEXO: sistema calcular

- INPUT DE DATOS :
- EXPANSION EN CANTIDAD DE AUTOS Y EN CANTIDAD DE HORAS SIMULADAS::::::::

```
function varargout = calcular(varargin)
% CALCULAR MATLAB code for calcular.fig
%      CALCULAR, by itself, creates a new CALCULAR or raises the existing
%      singleton*.
%
%      H = CALCULAR returns the handle to a new CALCULAR or the handle to
%      the existing singleton*.
%
%      CALCULAR('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in CALCULAR.M with the given input arguments.
%
%      CALCULAR('Property','Value',...) creates a new CALCULAR or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before calcular_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to calcular_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help calcular

% Last Modified by GUIDE v2.5 11-Jun-2014 00:43:26

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @calcular_OpeningFcn, ...
                   'gui_OutputFcn',  @calcular_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before calcular is made visible.
function calcular_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to calcular (see VARARGIN)

% Choose default command line output for calcular
handles.output = hObject;
```

```
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes calcular wait for user response (see UIRESUME)
% uiwait(handles.calcular);


% --- Outputs from this function are returned to the command line.
function varargout = calcular_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



function numAutos_Callback(hObject, eventdata, handles)
% hObject    handle to numAutos (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of numAutos as text
%        str2double(get(hObject,'String')) returns contents of numAutos as a double

numAutos = str2double(get(hObject, 'String'));
% --- Executes during object creation, after setting all properties.
function numAutos_CreateFcn(hObject, eventdata, handles)
% hObject    handle to numAutos (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function diasSim_Callback(hObject, eventdata, handles)
% hObject    handle to diasSim (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of diasSim as text
%        str2double(get(hObject,'String')) returns contents of diasSim as a double

numHoras= str2double(get(hObject, 'String'));
% --- Executes during object creation, after setting all properties.
function diasSim_CreateFcn(hObject, eventdata, handles)
% hObject    handle to diasSim (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function tecDisferentes_Callback(hObject, eventdata, handles)
% hObject    handle to tecDisferentes (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB


% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tecDisferentes as text
%        str2double(get(hObject,'String')) returns contents of tecDisferentes as a double

pasosDeCarga= str2double(get(hObject, 'String'));
% --- Executes during object creation, after setting all properties.
function tecDisferentes_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tecDisferentes (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end




function numeroDeAmpliacion_A_Callback(hObject, eventdata, handles)
% hObject    handle to numeroDeAmpliacion_A (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of numeroDeAmpliacion_A as text
%        str2double(get(hObject,'String')) returns contents of numeroDeAmpliacion_A as a double

numeroDeAmpliacion_A= str2double(get(hObject, 'String'));
% --- Executes during object creation, after setting all properties.
function numeroDeAmpliacion_A_CreateFcn(hObject, eventdata, handles)
% hObject    handle to numeroDeAmpliacion_A (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function numeroDeAmpliacion_H_Callback(hObject, eventdata, handles)
% hObject    handle to numeroDeAmpliacion_H (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of numeroDeAmpliacion_H as text
%        str2double(get(hObject,'String')) returns contents of numeroDeAmpliacion_H as a double

numeroDeAmpliacion_H= str2double(get(hObject, 'String'));
% --- Executes during object creation, after setting all properties.
function numeroDeAmpliacion_H_CreateFcn(hObject, eventdata, handles)
% hObject    handle to numeroDeAmpliacion_H (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes on button press in calcular.
function Calcular_Callback(hObject, eventdata, handles)
% hObject    handle to calcular (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% hObject    handle to demandasAutos_Horas_MarkovCondicionado (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
numeroDeAmpliacion_H=str2double(get(handles.numeroDeAmpliacion_H,'string'));
numeroDeAmpliacion_A=str2double(get(handles.numeroDeAmpliacion_A,'string'));

P1=str2num(get(handles.P1,'string'));
P2=str2num(get(handles.P2,'string'));
P3=str2num(get(handles.P3,'string'));

Mat=str2num(get(handles.Mat,'string'));

numAutos=str2double(get(handles.numAutos,'string'));
diasSim=str2double(get(handles.diasSim,'string'));


tecDisferentes=str2num(get(handles.tecDisferentes,'string'));
tecEE=str2num(get(handles.tecEE,'string'));
```

INPUT DE DATOS :

```
% % DATOS de entrada para simular ::::::::::::::::::::::::::::::::::
% numAutos=100;
% diasSim=5;

%%::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::


%EXPANSION DE LOS DATOS
% numeroDeAmpliacion_H=2;   % en HORAS
% numeroDeAmpliacion_A=600;  % en CANTIDAD DE AUTOS



% BATERIAS
%
% %tecnologias
% tecEE=[24 24 25 16];  % KWh de bateria , en este caso se consideran las marcas: Nissan Leaf ,
Renault Fluence, Think City, Mitsubishi i-Miev
% tecDisferentes=[6 8 6 7];   %horas de carga (pasos de carga de cada marca distinta)


%Modelo                 (kWh)   Autonomía (km)     kWhBatería/100km
%_____________________________________________________________________
% Mega e-City           9      |     100      |       9
% Reva L-ion            11     |     120      |       9,17
% Think City            25     |     200      |       12,50
% Mitsubishi i-Miev     16     |     130      |       12,31
```

```
% Citröen C-Zero        16     |     130      |       12,31
% Renault Fluence ZE    24     |     175      |       13,71
% Nissan Leaf           24     |     160      |       15,00
% Tesla Roadster42      42     |     257      |       16,34


% Mat=[.25 .25 .25 .25;.25 .25 .25 .25;.25 .25 .25 .25;.25 .25 .25 .25]; %matriz de
probabilidades para el Markov del uso de una determinada tecnologia (tipo de bateria)
% %los datos en esta matriz podrian ser mas precisos basados en
% %estadisticas reales de las cantidades de diferentes tecnologias en el mercado (ventas, por
ejemplo..)









%Matrices de transicion de Markov sin condiciones
%
% P1=[75 5 10 10;55 5 20 20;55 5 20 20;65 5 20 10]/100;%horario de la madrugada
% P2=[43 3 34 20;33  13 34 20;33  3 24 40;33  13 34 20]/100;%horario de 6 a 18
% P3=[30 20 30 20;35 25 20 20;25 35 40 20;20 30 30 20]/100;%horario de la noche
% %los datos de estas matrices podrian ser mas precisos basados en
%estadisticas reales de cuantos autos circulan en determinado momento del
%dia, etc..
%%_____________________________________________________________________



numHoras=diasSim*24;
bat = rand(numAutos,numHoras)*min(tecEE); % aca se guarde el registro de todos los autos hora a
hora
cadena1 = zeros(numAutos,numHoras); % cadena de Markov no condicionada por cada auto
cadena1(:,1)=1; % parto siempre del estado 1 en todos los autos


cadenaMCond = zeros(numAutos,numHoras);
Demanda_EECargaConstante=zeros(numAutos,numHoras);
Kilometraje=zeros(numAutos,numHoras);
cadenaMCond(:,1)=1; % parto siempre del estado 1 en todos los autos


bw=waitbar(0);



for auto=1:numAutos


    % BATERIAS por cada auto

 sorteoTec=hmmgenerate(1,Mat,eye(size(Mat))); % le asigno un tipo de tecnologia a cada auto
 % se asume un mercado de 4 tecnologias diferentes (ver cuadro)

 switch sorteoTec

     case 1  % ver la tercer columna del cuadro:kWhBatería/100km
         coef=15;
     case 2
         coef=13.71;
     case 3
         coef=12.5;
     case 4
```

```
         coef=12.31;
 end


 batTope=tecEE(sorteoTec);  %EE
 pasosDeCarga=tecDisferentes(sorteoTec); % modelamos diferentes cambios en los pasos de carga
para distintos autos
 deltaBat=batTope/pasosDeCarga;
 batMin=deltaBat; % se considera que "deltaBat" es el minimo para el funcionamiento (entregando o
andando)


for h=2:numHoras % ver de hacerlo con un while 7 dias , 7 vueltas al reloj de 24 horas


    faseDiaria=h*pi/12;  %modulo24









 if cos(faseDiaria)<0.01  % distinto de cero para evitar error de truncamiento del


%
        distribucion_paso = P2(cadena1(auto,h-1),:); % P2 es la matriz del poste horario 2


        distribucion_acumulada = cumsum(distribucion_paso);
        r = rand();
        cadena1(auto,h) = find(distribucion_acumulada>r,1);



        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%CONDICION


        c2=and(batMin<bat(auto,h),bat(auto,h)<=batTope);
        c3=not(bat(auto,h)>=batTope);

        PC2=[P2(1,1)*c3 P2(1,2)*c2 P2(1,3)*c2 1-(P2(1,1)*c3+P2(1,2)*c2+P2(1,3)*c2);P2(2,1)*c3
P2(2,2)*c2 P2(2,3)*c2 1-(P2(2,1)*c3+P2(2,2)*c2+P2(2,3)*c2)...
            ;P2(3,1)*c3 P2(3,2)*c2 P2(3,3)*c2 1-(P2(3,1)*c3+P2(3,2)*c2+P2(3,3)*c2);P2(4,1)*c3
P2(4,2)*c2 P2(4,3)*c2 1-(P2(4,1)*c3+P2(4,2)*c2+P2(4,3)*c2)];%horario de 6 a 18

        distribucion_pasoC = PC2(cadenaMCond(auto,h-1),:); % P2 es la matriz del poste horario 2
        distribucion_acumulada = cumsum(distribucion_pasoC);
        r = rand();
        cadenaMCond(auto,h) = find(distribucion_acumulada>r,1);

       if and(cadenaMCond(auto,h)==1,bat(auto,h)+ deltaBat<batTope)  %en el estado 1 demandando
deltaBat por hora al sistema
            Demanda_EECargaConstante(auto,h)=deltaBat;
           bat(auto,h)=bat(auto,h)+ deltaBat; % en el estado 1 carga deltaBat por hora
        else if (cadenaMCond(auto,h)==2)%en el estado 2 entregando deltaBat por hora al sistema
               Demanda_EECargaConstante(auto,h)=-deltaBat;
               bat(auto,h)=bat(auto,h)- deltaBat;
            else if (cadenaMCond(auto,h)==3)%en el estado 3 el auto esta andando
                    Kilometraje(auto,h)=deltaBat*100/coef;  %dado el coeficiente de
(kWhBatería/100km) y la demanda de Energia en KWh del paso: deltaBat
                    bat(auto,h)=bat(auto,h)- deltaBat; % no interactua con sistema al estar
andando pero si se le consume deltaBat por hora a la bateria
```

```
                end
            end
        end



    else if sin(faseDiaria)>0.01


        distribucion_paso = P1(cadena1(auto,h-1),:); % P1 es la matriz del poste horario 1 para
el VERANO
        distribucion_acumulada = cumsum(distribucion_paso);
        r = rand();
        cadena1(auto,h) = find(distribucion_acumulada>r,1);



        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%CONDICION


        c2=double(and(batMin<bat(auto,h),bat(auto,h)<=batTope));
        c3=double(not(bat(auto,h)>=batTope));







       PC1=[P1(1,1)*c3 P1(1,2)*c2 P1(1,3)*c2 1-(P1(1,1)*c3+P1(1,2)*c2+P1(1,3)*c2);P1(2,1)*c3
P1(2,2)*c2 P1(2,3)*c2 1-(P1(2,1)*c3+P1(2,2)*c2+P1(2,3)*c2)...
            ;P1(3,1)*c3 P1(3,2)*c2 P1(3,3)*c2 1-(P1(3,1)*c3+P1(3,2)*c2+P1(3,3)*c2);P1(4,1)*c3
P1(4,2)*c2 P1(4,3)*c2 1-(P1(4,1)*c3+P1(4,2)*c2+P1(4,3)*c2)];%horario de 6 a 18

        distribucion_pasoC = PC1(cadenaMCond(auto,h-1),:); % P1 es la matriz del poste horario 1
        distribucion_acumulada = cumsum(distribucion_pasoC);
        r = rand();
        cadenaMCond(auto,h) = find(distribucion_acumulada>r,1);

          if and(cadenaMCond(auto,h)==1,bat(auto,h)+ deltaBat<batTope)  %en el estado 1
demandando deltaBat por hora al sistema
            Demanda_EECargaConstante(auto,h)=deltaBat;
           bat(auto,h)=bat(auto,h)+ deltaBat; % en el estado 1 carga deltaBat por hora
        else if (cadenaMCond(auto,h)==2)%en el estado 2 entregando deltaBat por hora al sistema
               Demanda_EECargaConstante(auto,h)=-deltaBat;
               bat(auto,h)=bat(auto,h)- deltaBat;
            else if (cadenaMCond(auto,h)==3)%en el estado 3 el auto esta andando
                    Kilometraje(auto,h)=deltaBat*100/coef;  %dado el coeficiente de
(kWhBatería/100km) y la demanda de Energia en KWh del paso: deltaBat
                    bat(auto,h)=bat(auto,h)- deltaBat; % no interactua con sistema al estar
andando pero si se le consume deltaBat por hora a la bateria
                end
            end
          end


        else



        distribucion_paso = P3(cadena1(auto,h-1),:); % P3 es la matriz del poste horario 3 para
el VERANO

        distribucion_acumulada = cumsum(distribucion_paso);
        r = rand();
```

```
        cadena1(auto,h) = find(distribucion_acumulada>r,1);


           %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%CONDICION


        c2=and(batMin<bat(auto,h),bat(auto,h)<=batTope);
        c3=not(bat(auto,h)>=batTope);


        PC3=[P3(1,1)*c3 P3(1,2)*c2 P3(1,3)*c2 1-(P3(1,1)*c3+P3(1,2)*c2+P3(1,3)*c2);P3(2,1)*c3
P3(2,2)*c2 P3(2,3)*c2 1-(P3(2,1)*c3+P3(2,2)*c2+P3(2,3)*c2)...
            ;P3(3,1)*c3 P3(3,2)*c2 P3(3,3)*c2 1-(P3(3,1)*c3+P3(3,2)*c2+P3(3,3)*c2);P3(4,1)*c3
P3(4,2)*c2 P3(4,3)*c2 1-(P3(4,1)*c3+P3(4,2)*c2+P3(4,3)*c2)];%horario de 6 a 18

        distribucion_pasoC = PC3(cadenaMCond(auto,h-1),:); % P3 es la matriz del poste horario 3
        distribucion_acumulada = cumsum(distribucion_pasoC);
        r = rand();
        cadenaMCond(auto,h) = find(distribucion_acumulada>r,1);

      if and(cadenaMCond(auto,h)==1,bat(auto,h)+ deltaBat<batTope) %en el estado 1 demandando
deltaBat por hora al sistema
           Demanda_EECargaConstante(auto,h)=deltaBat;
           bat(auto,h)=bat(auto,h)+ deltaBat; % en el estado 1 carga deltaBat por hora
        else if (cadenaMCond(auto,h)==2)%en el estado 2 entregando deltaBat por hora al sistema
               Demanda_EECargaConstante(auto,h)=-deltaBat;
               bat(auto,h)=bat(auto,h)- deltaBat;
            else if (cadenaMCond(auto,h)==3)%en el estado 3 el auto esta andando




bat(auto,h)=bat(auto,h)- deltaBat; % no interactua con sistema al estar andando pero si se le
consume deltaBat por hora a la bateria
                    Kilometraje(auto,h)=deltaBat*100/coef;  %dado el coeficiente de
(kWhBatería/100km) y la demanda de Energia en KWh del paso: deltaBat
                end
            end
       end

        end

    end

end %fin horas


bw=waitbar(auto/numAutos,bw,strcat('@_MArkovCondicionado....Auto:',num2str(auto),' de
',num2str(numAutos)));

end %fin autos
```

EXPANSION EN CANTIDAD DE AUTOS Y EN CANTIDAD DE HORAS SIMULADAS::::::::

```
%::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::


%// ETAPA PARA ESCALAR LOS DATOS EN CANTIDAD DE AUTOS_____________________


v=sum(Demanda_EECargaConstante);
```

```
w=sum(Kilometraje);
```

```
[cidx, ctrs] = kmeans(v,3);  %busco 3 clusters de datos en el arreglo de suma  en horas
[zidx, ztrs] = kmeans(w,3);  %busco 3 clusters de datos en el arreglo de suma  en horas
%cidx guarda la cadena de calores por hora luego de la clusterizacion
%ctrs guarda los valores representativos de cada cluster

Demanda_EECargaConstante_A=zeros(1,length(cidx));
Kilometraje_A=zeros(1,length(zidx));

for i=1:length(cidx)

    if cidx(i)==1
  Demanda_EECargaConstante_A(i)=numeroDeAmpliacion_A*max(ctrs(1),v(i));
  Kilometraje_A(i)=numeroDeAmpliacion_A*max(ztrs(1),w(i));
    else if cidx(i)==2
   Demanda_EECargaConstante_A(i)=numeroDeAmpliacion_A*max(ctrs(2),v(i));
   Kilometraje_A(i)=numeroDeAmpliacion_A*max(ztrs(2),w(i));
        else
       Demanda_EECargaConstante_A(i)=numeroDeAmpliacion_A*min(ctrs(3),v(i));
       Kilometraje_A(i)=numeroDeAmpliacion_A*min(ztrs(3),w(i));
        end
    end

end

%_____________________________________________________________________

%// ETAPA PARA EXPANDIR LOS DATOS EN HORAS_____________________________



Demanda_EECargaConstanteAmpl=[];
Kilometraje_Ampl=[];
for i=1:numeroDeAmpliacion_H



Demanda_EECargaConstanteAmpl=cat(2,Demanda_EECargaConstanteAmpl,Demanda_EECargaConstante_A);
Kilometraje_Ampl=cat(2,Kilometraje_Ampl,Kilometraje_A);
end

%_____________________________________________________________________


%GRAFICAS Y SALIDA DE DATOS EN XLS
close(bw);
plot(sum(Demanda_EECargaConstante));title(strcat('demandas de EE de
los',num2str(numeroDeAmpliacion_A*numAutos),' autos por cada
hora.'));xlabel('hora');ylabel('demandas KWh');
hold on
plot(Demanda_EECargaConstanteAmpl,'r');
hold on
plot(Demanda_EECargaConstante_A,'k')
figure
plot(sort(sum(Demanda_EECargaConstante)));title(strcat('monotona de demandas de EE de
los',num2str(numeroDeAmpliacion_A*numAutos)));xlabel('indices');ylabel('demandas KWh');
hold on
plot(sort(Demanda_EECargaConstanteAmpl),'r');
hold on
plot(sort(Demanda_EECargaConstante_A),'k')

% xlswrite('autoXLS1_MarkCond.xls',(1:length(Demanda_EECargaConstanteAmpl))','inputSIMSEE');
% xlswrite('autoXLS1_MarkCond.xls',Demanda_EECargaConstanteAmpl','inputSIMSEE','B1');
```

```
horaInDia=1;
horaFinDia=24;

dias=floor(numHoras*numeroDeAmpliacion_H/24);
demandas=zeros(1,dias);
KilometrajeD=zeros(1,dias);
for dia=1:dias

demandas(dia)=sum(Demanda_EECargaConstanteAmpl(horaInDia:horaFinDia));
KilometrajeD(dia)=sum(Kilometraje_Ampl(horaInDia:horaFinDia));
horaInDia=horaInDia+24;
horaFinDia=horaFinDia+24;

end


% xlswrite('autoXLS1_MarkCond_DIASMooth.xls',(1:length(demandas))','inputSIMSEE');
% xlswrite('autoXLS1_MarkCond_DIASMooth.xls',round(smooth(demandas))','inputSIMSEE','B1');
%guardo XLs el smooth de las demandas diarias (filtro picos indeseados, pasa bajos)
xlswrite('autoXLS1_MarkCond_DIAS.xls',(1:length(demandas))','inputSIMSEE');
xlswrite('autoXLS1_MarkCond_DIAS.xls',round(demandas)','inputSIMSEE','B1'); %guardo XLS completo
señal con pico de ruido
figure
plot(demandas/1000);title(strcat('demandas diarias de EE de
los',num2str(numeroDeAmpliacion_A*numAutos)));xlabel('dia');ylabel('demandas MWh'); %graficada en
MWh
figure
plot((KilometrajeD),'r');title(strcat('Kilometros recorridos por
los',num2str(numeroDeAmpliacion_A*numAutos),' autos por cada dia.'));xlabel('dia');ylabel('Km');
function P1_Callback(hObject, eventdata, handles)
% hObject    handle to P1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of P1 as text
%        str2double(get(hObject,'String')) returns contents of P1 as a double
P1= str2double(get(hObject, 'String'));

% --- Executes during object creation, after setting all properties.



function P1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to P1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function P2_Callback(hObject, eventdata, handles)
% hObject    handle to P2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of P2 as text
%        str2double(get(hObject,'String')) returns contents of P2 as a double
P2= str2double(get(hObject, 'String'));
```

```
% --- Executes during object creation, after setting all properties.
function P2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to P2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function P3_Callback(hObject, eventdata, handles)
% hObject    handle to P3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of P3 as text
%        str2double(get(hObject,'String')) returns contents of P3 as a double
P3= str2double(get(hObject, 'String'));

% --- Executes during object creation, after setting all properties.
function P3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to P3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function Mat_Callback(hObject, eventdata, handles)
% hObject    handle to Mat (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Mat as text
%        str2double(get(hObject,'String')) returns contents of Mat as a double




Mat= str2double(get(hObject, 'String'));

% --- Executes during object creation, after setting all properties.
function Mat_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Mat (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function tecEE_Callback(hObject, eventdata, handles)
% hObject    handle to tecEE (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tecEE as text
%        str2double(get(hObject,'String')) returns contents of tecEE as a double
tecEE= str2double(get(hObject, 'String'));

% --- Executes during object creation, after setting all properties.
function tecEE_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tecEE (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes during object creation, after setting all properties.
function plot1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to plot1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: place code in OpeningFcn to populate plot1

function figure1_ResizeFcn(hObject, eventdata, handles)
```