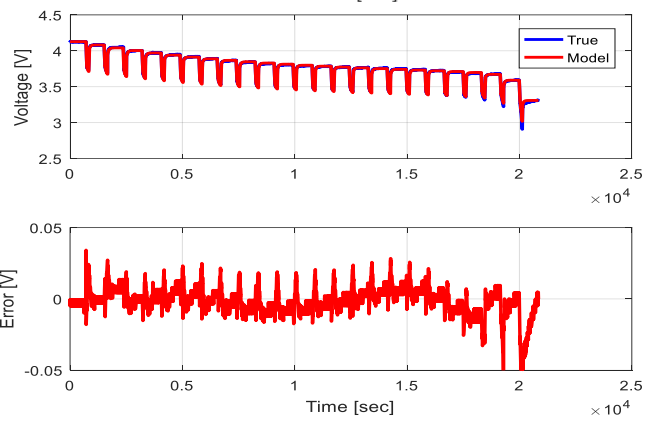
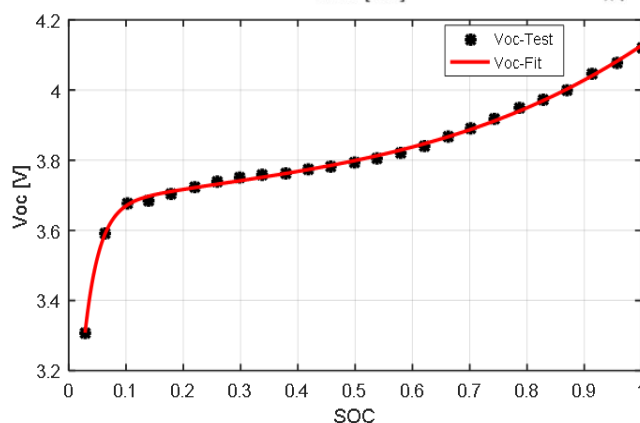
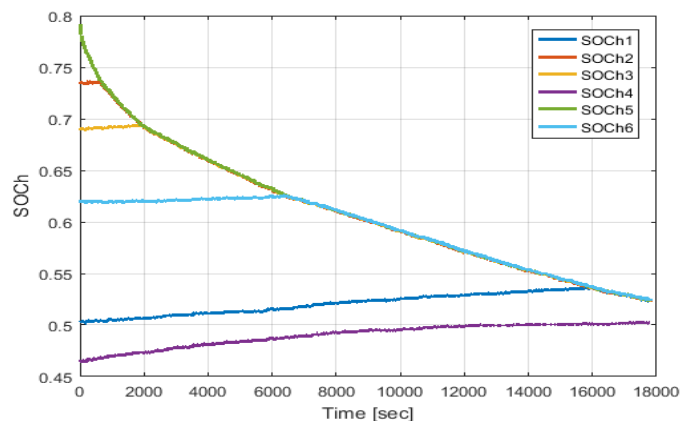
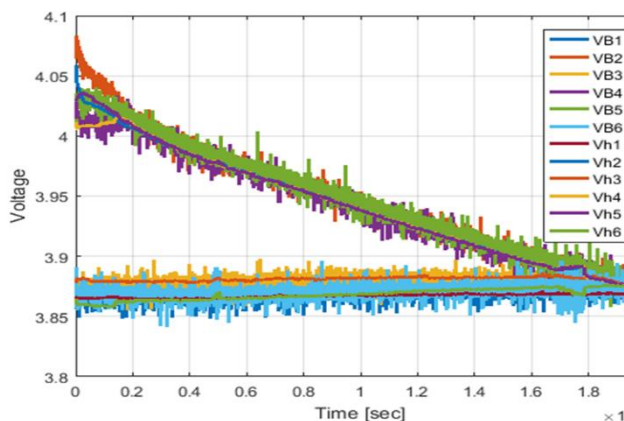


طراحی و ساخت سیستم مدیریت باتری فعال و غیر فعال مبتنی بر تخمین سطح شارژ با روتگر فیلتر کالمن توسعه یافته و مد لغزشی



مدیر پروژه: علی سلامتی / امید بزاز
مجری: ادوارد غریبیان / حمیدرضا خالصی

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

طراحی و ساخت سیستم مدیریت باتری فعال و غیر فعال مبتنی بر تخمین سطح شارژ با رایتگر فیلتر کالمن توسعه یافته و مد لغزشی

مدیر پروژه:
علی سلامتی / امید بزاز

همکاران پروژه:
مهدی قلیزاده

مجری:
ادوارد غریبیان / حمیدرضا خالصی

انتشارات پژوهشگاه نیرو
۱۴۰۰

کد گزارش: NRI-ER-703-PECPN12-00-L

DOI: 10.30503/nripress.2020.305

ایران. وزارت نیرو. پژوهشگاه نیرو. گروه پژوهشی پایش و کنترل نیروگاه
طراحی و ساخت سیستم مدیریت باتری فعال و غیر فعال مبتنی بر تخمین سطح شارژ با
رؤیتگر فیلتر کالمن توسعه یافته و مد لغزشی / مدیر پروژه: امید بزاز. کارفرما: پژوهشگاه نیرو،
۱۴۰۰.

۱۶۹ ص: مصور، نمودار، جدول

۱. سلامتی، علی، همکار پروژه. ۲. غریبیان، ادوارد، مجری. ۳. خالصی، حمیدرضا،
مجری.



انتشارات پژوهشگاه نیرو
گروه پژوهشی پایش و کنترل نیروگاه

کلیه حقوق قانونی این اثر متعلق به پژوهشگاه نیرو است.

نشانی: تهران، شهرک غرب، انتهای بلوار شهید دادمان، پژوهشگاه نیرو، کد پستی: ۱۴۶۸۶۱۳۱۱۳

تلفن: ۸۸۰۷۹۴۰۱-۹

نمابر: ۸۸۰۷۸۲۹۶

وبگاه: www.nri.ac.ir

پست الکترونیک انتشارات: publications@nri.ac.ir

شماره:

بسمه تعالی

تاریخ:

پیوست:



فرم تعهدنامه اصالت اثر

اینجانب امید بزاز متعهد می‌شوم که مطالب مندرج در این گزارش حاصل کار پژوهشی اینجانب و همکاران به شرح زیر است و به دستاوردهای پژوهشی دیگران که در این پژوهش از آن استفاده شده است، مطابق مقررات ارجاع و در فهرست منابع و مأخذ ذکر گردیده است. کلیه حقوق مادی و معنوی این اثر متعلق به پژوهشگاه نیرو است.

همکاران :

۱. مهدی قلیزاده

پیشگفتار

سیستم‌های ذخیره انرژی و از آن جمله باتری‌های الکتروشیمیایی در بسیاری از سیستم‌های الکتریکی از اهمیت زیادی برخوردارند، چراکه انرژی شیمیایی ذخیره شده در آن‌ها می‌تواند به انرژی الکتریکی تبدیل شده و در مواقع نیاز به سیستم‌های الکتریکی تحویل داده شود. این سیستم‌ها می‌توانند به عنوان مولد اصلی انرژی در کاربردهایی همچون خودرو برقی و یا به عنوان منبع انرژی پشتیبان در سایر سیستم‌های الکترومکانیکی به کار روند. به عنوان مثال باتری در توربین‌های بادی، جزئی مهم از سیستم حفاظت محسوب می‌شود بگونه‌ای که چنانچه سطح شارژ باتری از یک حد معین کمتر شود، فرمان توقف صادر می‌شود. در اینجا باتری وظیفه تامین انرژی لازم برای عملکرد سیستم ترمز آیرودینامیکی و ترمز مکانیکی را در مواقع قطع احتمالی شبکه دارد تا از سرعت‌گیری توربین در اثر وزش باد و آسیب‌های مکانیکی متعاقب آن جلوگیری کند.

طول عمر باتری یکی از فاکتورهای اصلی محدود کننده توسعه انرژی‌های نو است. علیرغم تحقیقات فراوانی که طی ۳۰ سال گذشته صورت گرفته، باز هم قیمت و طول عمر باتری به گونه‌ای است که مهم‌ترین عامل بازدارنده تولید خودروهای الکتریکی و الکتریکی-ترکیبی می‌باشد. باتری‌های لیتیوم-یون عمر مفید بیشتری در مقایسه با سایر باتری‌ها نظیر اسید-سرب دارند و می‌توانند طی بیش از ۱۰۰۰ سیکل شارژ و دشارژ کارایی مناسبی را از خود به نمایش بگذارند. همچنین یکی از موانع عمده بر سر راه گسترش استفاده از انرژی‌های نو، موضوع زمانبر بودن شارژ مجدد این باتری‌ها است. بنابراین طراحی روشی برای شارژ خودروهای ترکیبی به گونه‌ای که دارای دو خصوصیت کاهش زمان شارژ و افزایش طول عمر باتری باشد، ضروری به نظر می‌رسد. به علت حساسیت‌های خاص باتری‌های لیتیوم-یون، شارژ و دشارژ آن‌ها بایستی توسط سیستم‌های مدیریت باتری کنترل گردد.

در اغلب کاربردها، سیستم باتری شامل یک باتری یا مجموعه‌ای از باتری‌ها و یک سیستم مدیریت باتری (BMS) است. یک BMS از سخت‌افزار و نرم‌افزاری تشکیل شده است که شارژ و دشارژ باتری را ضمن تضمین عملکرد ایمن و قابل اعتماد، کنترل می‌کند. مدیریت تعادل شارژ و مدیریت گرمایی از جمله مواردی است که بر عهده BMS است. با افزایش عرضه خودروهای الکتریکی (EV) به بازارهای جهانی و اهمیت باتری استفاده شده در این خودروها، نیاز به یک سیستم مدیریت باتری بیشتر احساس می‌شود. از جمله اجزاء اصلی سیستم مدیریت باتری می‌توان به کنترل کننده شارژ باتری اشاره کرد. یک BMS خبره با مشاهده حالت‌ها و پایش پارامترهای فیزیکی باتری در حین استهلاک آن و با مدیریت مناسب فرایند شارژ و دشارژ، تأثیر بسزایی در افزایش طول عمر و بهبود عملکرد باتری دارد.

یکی از پارامترهای مهم باتری، سطح شارژ باتری (SOC) است. سطح شارژ باتری عبارت است از نسبت ظرفیت باقیمانده در یک باتری به ظرفیت کل آن و در واقع شارژ موجود در الکتروود باتری را که بر حسب غلظت یون موردنظر محاسبه شده است، نشان می‌دهد. این پارامتر توسط الگوریتم‌های سیستم مدیریت باتری جهت تخمین انرژی ذخیره شده در مجموعه باتری و همچنین جهت بالانس کردن سلول‌های آن بکار گرفته می‌شود. از طرفی، ظرفیت سلول با گذشت عمر آن به تدریج کاهش پیدا می‌کند. این کاهش ظرفیت که علاوه بر میزان عمر باتری به دمای محیط و سرعت دشارژ باتری نیز مرتبط است، با معیار کاهش سطح سلامت باتری (SOH) سنجیده می‌شود. به عنوان مثال، ممکن است در یک سلول کهنه در حالت شارژ کامل، SOC فقط ۸۰ درصد باشد. بنابراین برای یک تخمین دقیق باید فاکتورهای کهنه شدن سلول و عوامل محیطی را نیز در محاسبات وارد کرد. اطلاع از SOH باتری این امکان را فراهم می‌سازد تا درباره عملکرد آینده باتری اظهار نظر کرد و از قابلیت آن برای تأمین توان بهترین استفاده را برد. مهم‌تر از همه، با دانستن انتهای عمر باتری، می‌توان در زمان

مناسب جایگزینی برای آن در نظر گرفت. برای مثال در خودروهای الکتریکی و الکتریکی ترکیبی (HEV)، اطلاع دقیق از SOC و SOH باتری، پیاده‌سازی استراتژی‌های مدیریت توان پیشرفته را میسر ساخته و به این ترتیب مزایایی همچون: کاهش نرخ مصرف سوخت، افزایش عمر باتری با نگهداری آن در یک محدوده SOC بهینه، کاهش وقوع خرابی، بهینه‌گی اندازه و هزینه باتری و ژنراتور، کاهش هزینه ضمانت باتری و بهبود ایمنی وسیله نقلیه با تعیین و تنظیم بهینه پارامترهای باتری مرتبط با ایمنی وسیله حاصل می‌شود.

همانطور که گفته شد، کنترل فرآیند شارژ و دشارژ باتری به گونه‌ای که سطح شارژ همواره در یک محدوده مشخص از قبل تعیین شده قرار داشته باشد، باعث افزایش چشمگیر عمر باتری و تعداد سیکل‌های شارژ و دشارژ آن می‌گردد. از آنجا که دانستن سطح شارژ باتری در هر لحظه برای کنترل سیستم باتری لازم است، نیاز به بکارگیری روش‌هایی است که با اندازه‌گیری جریان و ولتاژ پایانه باتری و استفاده از دینامیک مدل شده باتری، انرژی داخلی آن را در حالت کار بطور پیوسته تخمین بزند. به این ترتیب با استفاده از این تخمین گر سطح شارژ و روش‌های کنترلی مناسب، می‌توان سطح شارژ باتری را به صورت مطلوب کنترل کرد.

استک‌های باتری در خودروهای هیبریدی الکتریکی معمولاً از تعداد زیادی باتری که بطور سری بهم پیوسته‌اند، جهت ایجاد ولتاژ ۳۰۰-۴۰۰ ولت DC تشکیل شده‌اند. وجود تفاوت‌هایی در خصوصیات شیمیایی سلول‌ها نظیر امپدانس داخلی و ظرفیت، عدم یکسانی دمای سلول‌های قرار گرفته درون یک استک و تفاوت‌های معمول در سیکل‌های شارژ و دشارژ، منجر به پدیده عدم تعادل شارژ بین سلول‌های مختلف یک استک باتری خواهد گشت. با توجه به اینکه سیستم کنترل شارژ باتری‌ها معمولاً با توجه به ولتاژ کل استک تصمیم‌گیری می‌کند، احتمال آنکه ولتاژ یک سلول مجزا در حین شارژ از حد بیشینه تعیین شده بالاتر رود و یا در حین دشارژ بیشتر از حد مجاز تخلیه گردد وجود دارد. اگر یک سلول لیتیوم یون بیش از اندازه شارژ گردد، فشار و دمای آن سلول به شدت بالا رفته و می‌تواند حتی منجر به انفجار سلول شود؛ درحالی‌که تخلیه بیش از حد سلول‌ها موجب کاهش شدید ظرفیت مفید آن‌ها می‌گردد. در صورت وقوع چنین رخدادی، استک باتری تأمین‌کننده انرژی خودرو می‌تواند با پدیده‌هایی نظیر از دست دادن ظرفیت یک یا چند سلول و یا حتی انفجار روبرو گردد. بنابراین استفاده از روشی که بتواند با پدیده‌ی عدم تعادل بین سلول‌های باتری مقابله کند بسیار مهم است.

از یک دیدگاه، روش‌های مختلف یکسانسازی شارژ سلول‌ها را می‌توان به دو دسته فعال و غیر فعال تقسیم نمود. تفاوت اساسی این دو گروه در آن است که در روش‌های یکسانسازی غیر فعال، شارژ اضافی سلول با ولتاژ بالاتر به گونه‌ای تلف می‌شود تا ولتاژ همه سلول‌های استک به یک سطح برسد. این در حالیست که در یکسانسازی فعال، شارژ اضافی سلول مورد نظر، به سایر سلول‌های با ولتاژ پایینتر انتقال داده می‌شود. در این یکسانسازی‌ها، انرژی اضافی سلول‌ها به جای اینکه تلف شود، به وسیله‌ی عناصر فعالی نظیر خازن، مبدل‌های dc به dc ایزوله و یا غیر ایزوله و یا ترکیبی از آن‌ها به سلول‌های کم شارژتر منتقل گشته و به این ترتیب باعث بالا رفتن بازدهی این روش‌ها خواهد شد.

در این پروژه قصد بر آن است تا یک سیستم جامع مدیریت استک‌های باتری به کار رفته در بخش Back-up نیروگاه‌ها و منابع تولید پراکنده به صورت جامع مورد طراحی، صحت‌سنجی و سپس پیاده‌سازی قرار گیرد. با توجه به تأثیر طراحی رویتگر برای تخمین سطح شارژ باتری‌ها به جای استفاده از ولتاژ سلول‌ها در فرایند کنترل، در این طرح نخست عملکرد تخمین‌گرهای مختلف از جمله فیلتر کالمن توسعه‌یافته و رویتگر مد لغزشی، برای تخمین سطح شارژ از طریق شبیه‌سازی مورد بررسی و مقایسه قرار می‌گیرد. در گام بعدی، برای طراحی یکسانساز شارژ به عنوان هسته اصلی سیستم مدیریت ذخیره‌ساز، روش‌های فعال (Active) و غیر فعال (Passive) مورد بررسی قرار می‌گیرند. فاز انتهایی پروژه مشتمل بر طراحی و پیاده‌سازی عملی الگوریتم تخمین مناسب برای هر دو روش یکسانسازی فعال و غیر فعال می‌باشد. با توجه به عدم وجود نمونه‌ی داخلی برای

سیستم‌های یاد شده و به خصوص سیستم مدیریت باتری فعال با قابلیت مازولار شدن قابل استفاده در زنجیره‌های مشتمل بر چندصد باتری، امکان ثبت محصول نهایی پروژه به عنوان اختراع نیز محفوظ می‌باشد.

انجام این پروژه در ۴ فاز صورت گرفت. در فاز اول با عنوان "استخراج مدل دینامیکی باتری و شناسایی پارامترهای آن" با مطالعه ویژگی‌های تکنولوژی‌های مختلف باتری، باتری‌های لیتیوم-یون به علت خصوصیات منحصر بفردی نظیر طول عمر، تعداد دفعات قابل شارژ و دشارژ، گستره ولتاژ قابل توان‌دهی و... به عنوان مبنای طراحی سیستم BMS تعیین می‌گردد. سپس با بررسی روش‌های مختلف مدل‌سازی باتری، یک مدل دینامیکی از باتری مبتنی استخراج می‌شود. در ادامه الگوریتم‌های شناسایی هر یک از پارامترهای دینامیکی مدل طراحی شد و در نرم‌افزار Matlab شبیه‌سازی می‌گردد.

فاز دوم با عنوان "طراحی و شبیه‌سازی روش‌های تخمین سطح شارژ مبتنی بر فیلتر کالمن توسعه یافته و ر‌ؤیتگر مد لغزشی" با بکارگیری روش‌های فیلتر کالمن توسعه یافته و ر‌ؤیتگر مد لغزشی به طراحی الگوریتم‌های تخمین سطح شارژ سلول با بکارگیری مدل دینامیکی باتری و استفاده از داده‌های ولتاژ و جریان و می‌پردازد و در ادامه با شبیه‌سازی در نرم‌افزار Matlab صحت روش‌های طراحی شده را بررسی می‌کند.

"طراحی الگوریتم‌های یکسان‌سازی سطح شارژ به روش‌های فعال و غیرفعال" عنوان فاز سوم است که در آن به بررسی روش‌های مختلف یکسان‌سازی سطح شارژ بصورت فعال و غیرفعال و انتخاب روش مناسب می‌پردازد. در فاز آخر پروژه با عنوان "پیاده‌سازی سخت‌افزاری و ساخت یک سیستم BMS آزمایشگاهی و تست الگوریتم‌های شناسایی و یکسان‌سازی" جهت پیاده‌سازی و اجرای عملی الگوریتم‌های طراحی شده در مراحل قبل، یک سیستم BMS با بردهای مجزا برای شناسایی و یکسان‌سازی طراحی می‌گردد. این سیستم قابلیت اتصال همزمان به ۶ سلول و اجرای پروسه یکسان‌سازی را بر روی آنها دارد. برای شناسایی پارامترها و برای تخمین سطح شارژ و اجرای پروسه یکسان‌سازی از داده‌های سطح شارژ و ولتاژ و جریان هر سلول استفاده می‌گردد. در نهایت با انجام آزمایش و بررسی نتایج عملی، عملکرد الگوریتم‌های طراحی شده مورد ارزیابی قرار می‌گیرد.

این پروژه با مدیریت آقای مهندس علی سلامتی آغاز شد و تا مرحله ساخت دستگاه پیش رفت. در ادامه آقای مهندس امید بزاز مدیریت آن را بر عهده گرفت و مراحل ارزیابی و صحت‌سنجی دستگاه و تحویل‌دهی آن به انجام رسید. همچنین جناب آقای مهندس حسام امیری مسئولیت نظارت را در این پروژه عهده‌دار شدند و آقای مهندس مهدی قلی‌زاده در طراحی الگوریتم‌ها و ساخت دستگاه همکاری داشتند

فهرست مطالب

عنوان	صفحه
۱- بررسی تحلیلی الگوریتم ها و کاربرد آن ها در روش های یکسان سازی فعال و غیر فعال	۱۰
۱-۱- مقدمه	۱۰
۱-۲- مقدمه ای بر باتری	۱۱
۱-۲-۱- تاریخچه باتری	۱۱
۱-۲-۲- تکنولوژی های باتری	۱۲
۱-۲-۳- پارامترهای باتری	۱۶
۱-۳- مدل سازی باتری و شناسایی پارامترهای باتری	۱۷
۱-۳-۱- مدل های الکتریکی باتری	۱۸
۱-۳-۲- شناسایی پارامترهای باتری	۲۷
۱-۳-۳- روش های موجود برای تخمین SOC	۳۳
۱-۳-۴- تخمین سطح شارژ با استفاده از فیلتر کالمن	۳۵
۱-۳-۵- تخمین سطح شارژ با استفاده از رؤیتگر مد لغزشی	۳۸
۱-۴- روش های یکسان سازی شارژ باتری	۴۰
۱-۴-۱- روشهای یکسان سازی شارژ غیر فعال	۴۱
۱-۴-۲- یکسان سازی شارژ فعال	۴۲
۱-۵- جمع بندی	۶۱
۱-۶- مراجع	۶۳
۲- طراحی و شبیه سازی روش های مورد بررسی	۷۰
۲-۱- مقدمه	۷۰
۲-۲- شبیه سازی الگوریتم های تخمین SOC در MATLAB	۷۰
۲-۲-۱- مدل سازی باتری	۷۱
۲-۲-۲- شبیه سازی الگوریتم EKF در محیط MATLAB جهت تخمین سطح شارژ	۷۲
۲-۲-۳- شبیه سازی الگوریتم SMO در MATLAB	۷۷
۲-۳- شبیه سازی یکسان سازی شارژ غیرفعال در MATLAB	۸۳
۲-۳-۱- شبیه سازی یکسان سازی غیرفعال مبتنی بر تخمین سطح شارژ با استفاده از SMO	۸۳
۲-۳-۲- شبیه سازی یکسان سازی غیرفعال مبتنی بر تخمین سطح شارژ با استفاده از EKF	۸۸
۲-۴- شبیه سازی یکسان سازی شارژ فعال در MATLAB	۹۱
۲-۴-۱- شبیه سازی یکسان سازی فعال مبتنی بر مبدل فلای یک با استفاده از تخمینگر SMO	۹۴
۲-۴-۲- شبیه سازی یکسان سازی فعال مبتنی بر تخمین سطح شارژ با استفاده از EKF	۱۰۳
۲-۵- جمع بندی	۱۰۶
۲-۶- مراجع	۱۰۷
3- طراحی سخت افزاری و پیاده سازی الگوریتم های یکسان سازی شارژ فعال و غیرفعال	۱۰۸
۳-۱- مقدمه	۱۰۸

۳-۲- طراحی سخت افزاری مدار تست شناسایی پارامترهای مدل باتری و پیاده سازی الگوریتم های شناسایی	۱۰۸
۳-۲-۱- اجزای تشکیل دهنده مدار استخراج داده تست جهت شناسایی پارامترهای باتری	۱۰۹
۳-۲-۲- برنامه استخراج داده تست شناسایی	۱۱۰
۳-۲-۳- برنامه شناسایی پارامترهای باتری	۱۱۷
۳-۳- برنامه نویسی و پیاده سازی عملی الگوریتم های یکسان سازی شارژ	۱۲۰
۳-۳-۱- یکسان سازی غیرفعال	۱۲۰
۳-۳-۲- یکسان سازی فعال	۱۲۱
۳-۳-۳- اجزای تشکیل دهنده مدار تست جهت یکسان سازی شارژ	۱۲۲
۳-۳-۴- پیاده سازی روش های یکسان سازی بر روی DSP	۱۲۳
۳-۳-۵- لحاظ کردن مدیریت دمایی	۱۴۲
۳-۴- جمع بندی	۱۴۳
۳-۵- مراجع	۱۴۴
۴- انجام آزمایش و ارزیابی نتایج	۱۴۵
۴-۱- مقدمه	۱۴۵
۴-۲- معرفی دستگاه یکسان ساز شارژ	۱۴۶
۴-۳- شناسایی پارامترهای باتری	۱۴۷
۴-۳-۱- روند شناسایی پارامترهای باتری	۱۴۸
۴-۳-۲- استخراج منحنی V_{OC} بر حسب SOC	۱۴۹
۴-۳-۳- تخمین پارامترها	۱۵۰
۴-۴- نتایج عملی یکسان سازی شارژ سلول های باتری	۱۵۳
۴-۴-۱- نحوه استفاده از مدار یکسان ساز شارژ	۱۵۳
۴-۴-۲- یکسان سازی غیرفعال با EKF	۱۵۵
۴-۴-۳- یکسان سازی غیرفعال با SMO	۱۵۶
۴-۴-۴- یکسان سازی فعال با EKF	۱۵۶
۴-۴-۵- یکسان سازی فعال با SMO	۱۵۷
۴-۵- جمع بندی	۱۵۹
پیوست	۱۶۰
۴-۶- مراجع	۱۶۱

فهرست شکل‌ها

عنوان.....	صفحه
شکل ۱-۱- شماتیک یک سلول لیتیم-یون [۲].....	۱۵
شکل ۱-۲- مدل الکتریکی بر مبنای تونن [۱].....	۱۹
شکل ۱-۳- مدل الکتریکی بر مبنای امیدانس [۱].....	۱۹
شکل ۱-۴- مدل الکتریکی بر مبنای زمان اجرا [۱].....	۲۰
شکل ۱-۵- یک مدل الکتریکی دقیق بر مبنای زمان اجرا از باتری [۱].....	۲۰
شکل ۱-۶- منحنی‌های مشخصه باتری؛ ظرفیت قابل استفاده باتری بر حسب (A) تعداد سیکل، (B) دما، (C) جریان، (D) زمان ذخیره‌سازی و (E) ولتاژ مدار باز بر حسب SOC و پاسخ گذرای باتری به جریان بار پله‌ای [۱].....	۲۱
شکل ۱-۷- مدل الکتریکی باتری برای استفاده در شبیه‌سازی HEV در یک دمای ثابت [۱۰].....	۲۲
شکل ۱-۸- مدل مدار معادل رندل [۲۵].....	۲۴
شکل ۱-۹- مدل انتخابی باتری از مرجع [۱].....	۲۸
شکل ۱-۱۰- منحنی V_{OC} بر حسب SOC.....	۳۱
شکل ۱-۱۱- ساختار مدل ARMAX.....	۳۲
شکل ۱-۱۲- مدار یکسان‌ساز غیر فعال بدون قابلیت کنترل [۱۰۵].....	۴۲
شکل ۱-۱۳- مدار یکسان‌ساز شارژ غیرفعال با قابلیت کنترل [۱۰۶].....	۴۲
شکل ۱-۱۴- مدار یکسان‌ساز خازنی تک‌شاخه [۱۰۸].....	۴۳
شکل ۱-۱۵- منطق کلیدزنی یکسان‌ساز خازنی تک‌شاخه [۱۰۹].....	۴۴
شکل ۱-۱۶- یکسان‌ساز خازنی دو شاخه [۱۰۹].....	۴۴
شکل ۱-۱۷- سیستم یکسان‌سازی مبتنی بر دایورترهای جریان دوطرفه [۱۱۰].....	۴۵
شکل ۱-۱۸- ساختار داخلی مبدل‌های دوطرفه [۱۱۰].....	۴۵
شکل ۱-۱۹- وضعیت مداری مبدل در حالتی که Q_j روشن و D_{j+1} خاموش می‌باشد. $(V_{Bj} > V_{Bj+1})$ [۱۱۰].....	۴۶
شکل ۱-۲۰- وضعیت مداری مبدل در حالتی که Q_j روشن می‌باشند. $(V_{Bj} > V_{Bj+1})$ [۱۱۰].....	۴۶
شکل ۱-۲۱- وضعیت مداری مبدل در حالتی که Q_j خاموش و D_{j+1} روشن می‌باشد. $(V_{Bj} > V_{Bj+1})$ [۱۱۰].....	۴۷
شکل ۱-۲۲- شکل موج سیگنال‌های مهم مدار مبدل در فواصل زمانی مختلف $(V_{Bj} > V_{Bj+1})$ [۱۱۰].....	۴۷
شکل ۱-۲۳- مبدل باک-بوست مورد استفاده در یکسان‌سازهای دوطرفه [۱۱۱].....	۴۸
شکل ۱-۲۴- مبدل باک-بوست QRZCS عادی [۱۱۱].....	۴۸
شکل ۱-۲۵- مبدل QRZCS اصلاح شده [۱۱۱].....	۴۹
شکل ۱-۲۶- وضعیت مدار مبدل QRZCS در حالت $V_{Bj} > V_{Bj+1}$ [۱۱۱].....	۴۹
شکل ۱-۲۷- وضعیت مدار مبدل QRZCS در حالت $V_{Bj} < V_{Bj+1}$ [۱۱۱].....	۵۰
شکل ۱-۲۸- وضعیت مدار در فاصله زمانی $t_0 < t < t_1$ [۱۱۱].....	۵۱
شکل ۱-۲۹- وضعیت مدار در فاصله زمانی $t_1 < t < t_2$ [۱۱۱].....	۵۱
شکل ۱-۳۰- وضعیت مدار در فاصله زمانی $t_2 < t < t_3$ [۱۱۱].....	۵۲
شکل ۱-۳۱- وضعیت مدار در فاصله زمانی $t_3 < t < t_4$ [۱۱۱].....	۵۲
شکل ۱-۳۲- وضعیت مدار در فاصله زمانی $t_4 < t < t_5$ [۱۱۱].....	۵۲
شکل ۱-۳۳- بلوک دیاگرام سیستم کنترل مبتنی بر منطق فازی [۱۱۲].....	۵۳
شکل ۱-۳۴- توابع عضویت مربوط به استراتژی فازی برای تنظیم جریان یکسان‌سازی [۱۱۲].....	۵۴
شکل ۱-۳۵- مقادیر جریان I_{BEC} تولیدی توسط کنترل‌کننده فازی به ازای V_d و V_B مختلف [۱۱۲].....	۵۵
شکل ۱-۳۶- رابطه درصد روشن بودن کلید و ولتاژهای V_d و V_B مختلف [۱۱۲].....	۵۵
شکل ۱-۳۷- فلوچارت نحوه تولید فرمان روشن و یا خاموش بودن کلیدهای مختلف [۱۱۲].....	۵۶

- شکل ۳۸-۱- یکسان سازی شارژ با استفاده از مبدل DC-DC فلای بک مرکزی [۱۱۳]..... ۵۶
- شکل ۳۹-۱- نمایش نحوه عملکرد پیاده سازی یکسان سازی از بالا [۱۱۴]..... ۵۷
- شکل ۴۰-۱- یکسان ساز با مبدل فلای بک مرکزی با سیم پیچ های ثانویه متعدد [۱۱۳]..... ۵۸
- شکل ۴۱-۱- یکسان ساز با مبدل فلای بک مرکزی با یک سیم پیچ ثانویه [۱۱۵]..... ۵۸
- شکل ۴۲-۱- وضعیت مدار یکسان ساز در حالتی که سلول دوم در حال دشارژ شدن است [۱۱۶]..... ۵۹
- شکل ۴۳-۱- مدار ساده شده یکسان ساز با مبدل فلای بک [۱۱۶]..... ۵۹
- شکل ۴۴-۱- وضعیت مدار یکسان سازی با مبدل فلای بک مرکزی در حالتی که دیود ثانویه هدایت می کند [۱۱۶]..... ۶۰
- شکل ۴۵-۱- وضعیت ولتاژها و جریان ها مختلف مدار یکسان ساز طی مراحل مختلف [۱۱۶]..... ۶۱
- شکل ۱-۲- مدل الکتریکی باتری ۷۱
- شکل ۲-۲- مدار یکسان ساز شارژ غیرفعال ۸۳
- شکل ۲-۳- SOC سلول ها در طی یکسان سازی غیرفعال با روش SMO ۸۷
- شکل ۲-۴- ولتاژ سلول ها در طی یکسان سازی غیرفعال با روش SMO ۸۸
- شکل ۲-۵- جریان سلول ها در طی یکسان سازی غیرفعال با روش SMO ۸۸
- شکل ۲-۶- SOC سلول ها در طی یکسان سازی غیرفعال با روش EKF ۸۹
- شکل ۲-۷- ولتاژ سلول ها در طی یکسان سازی غیرفعال با روش EKF ۹۰
- شکل ۲-۸- جریان سلول ها در طی یکسان سازی غیرفعال با روش EKF ۹۰
- شکل ۲-۹- یکسان ساز با مبدل فلای بک مرکزی با سیم پیچ های ثانویه متعدد ۹۲
- شکل ۲-۱۰- وضعیت مدار یکسان ساز در حالتی که کلید S1 روشن می باشد ۹۲
- شکل ۲-۱۱- مدار ساده شده یکسان ساز با مبدل فلای بک ۹۳
- شکل ۲-۱۲- وضعیت ولتاژها و جریان ها مختلف مدار یکسان ساز طی مراحل مختلف ۹۴
- شکل ۲-۱۳- SOC سلول ها در طی یکسان سازی فعال با روش SMO ۱۰۲
- شکل ۲-۱۴- ولتاژ سلول ها در طی یکسان سازی فعال با روش SMO ۱۰۲
- شکل ۲-۱۵- جریان سلول ها در طی یکسان سازی فعال با روش SMO ۱۰۳
- شکل ۲-۱۶- میانگین ولتاژ سلول ها در بازه های سوئیچینگ در طی یکسان سازی فعال با روش SMO ۱۰۳
- شکل ۲-۱۷- SOC سلول ها در طی یکسان سازی فعال با روش EKF ۱۰۴
- شکل ۲-۱۸- ولتاژ سلول ها در طی یکسان سازی فعال با روش EKF ۱۰۴
- شکل ۲-۱۹- جریان سلول ها در طی یکسان سازی فعال با روش EKF ۱۰۵
- شکل ۲-۲۰- میانگین ولتاژ سلول ها در بازه های سوئیچینگ در طی یکسان سازی فعال با روش EKF ۱۰۵
- شکل ۳-۱- مدار تست جهت شناسایی پارامترها ۱۰۹
- شکل ۳-۲- مدار یکسان ساز شارژ غیرفعال ۱۲۱
- شکل ۳-۳- یکسان ساز با مبدل فلای بک مرکزی با سیم پیچ های ثانویه متعدد ۱۲۲
- شکل ۳-۴- مدار تست جهت یکسان سازی شارژ ۱۲۲
- شکل ۴-۱- دستگاه یکسان ساز شارژ ۱۴۶
- شکل ۴-۲- نمای جلوی محفظه (شکل بالا) و نمای پشت (شکل پایین) ۱۴۷
- شکل ۴-۳- مدل الکتریکی باتری ۱۴۷
- شکل ۴-۴- ولتاژ و جریان اندازه گیری شده به همراه نقاط مربوط به VOC ۱۴۹
- شکل ۴-۵- SOC محاسبه شده باتری به روش شمارش آمپر ساعت ۱۵۰
- شکل ۴-۶- منحنی VOC بر حسب SOC ۱۵۰
- شکل ۴-۷- جریان و ولتاژ استفاده شده برای محاسبه R_0 ۱۵۱
- شکل ۴-۸- خروجی اندازه گیری شده در مقایسه با خروجی مدل ۱۵۲
- شکل ۴-۹- برنامه DSP در نرم افزار CODECOMPOSER ۱۵۴
- شکل ۴-۱۰- ولتاژ سلولها در یکسان سازی غیرفعال با EKF ۱۵۵

۱۵۵.....	شکل ۴-۱۱- سطح شارژ سلولها در یکسان سازی غیرفعال با EKF
۱۵۶.....	شکل ۴-۱۲- ولتاژ سلولها در یکسان سازی غیرفعال با SMO
۱۵۶.....	شکل ۴-۱۳- سطح شارژ سلولها در یکسان سازی غیرفعال با SMO
۱۵۷.....	شکل ۴-۱۴- ولتاژ سلولها در یکسان سازی فعال با EKF
۱۵۷.....	شکل ۴-۱۵- سطح شارژ سلولها در یکسان سازی فعال با EKF
۱۵۸.....	شکل ۴-۱۶- ولتاژ سلولها در یکسان سازی فعال با SMO
۱۵۸.....	شکل ۴-۱۷- سطح شارژ سلولها در یکسان سازی فعال با SMO

فهرست جدول‌ها

عنوان.....	صفحه
جدول ۱-۳- خلاصه الگوریتم EKF [۲۶ و ۵۴].....	۳۶
جدول ۲-۳- روش‌های موجود برای تخمین SOC [۳۳ و ۶۴].....	۳۹
جدول ۱-۴- قواعد فازی استفاده شده در ماشین استنتاج برای متغیرهای زبانی [۱۱۲].....	۵۴
جدول ۱-۱- خلاصه الگوریتم EKF.....	۷۳
جدول ۱-۲- پارامترهای مدل باتری و مقاومت شنت مدار یکسان‌ساز.....	۸۴
جدول ۱-۲- متغیرهای جهانی به کار رفته در تابع اصلی (MAIN).....	۱۲۵
جدول ۲-۲- متغیرهای جهانی به کار رفته در MAIN و تخمینگرها.....	۱۲۵
جدول ۳-۲- پارامترهای رؤیتگر SMO.....	۱۲۶
جدول ۴-۲- پارامترهای تخمینگر EKF.....	۱۲۶
جدول ۵-۲- متغیرهای سلول شماره ۱ در الگوریتم SMO.....	۱۳۴
جدول ۶-۲- متغیرهای سلول شماره ۱ در الگوریتم EKF.....	۱۳۷
جدول ۱-۲- پارامترهای مدل باتری.....	۱۵۲
جدول ۱-۵- مشخصات پیک باتری مورد تست در پروژه.....	۱۶۰
جدول ۲-۵- مشخصات فنی محصول BMS تحویل داده شده.....	۱۶۰

۱- بررسی تحلیلی الگوریتم‌ها و کاربرد آن‌ها در روش‌های یکسان‌سازی فعال و غیر فعال

۱-۱- مقدمه

استک‌های باتری در خودروهای برقی و نیروگاه‌های تجدیدپذیر معمولاً از تعداد زیادی باتری که بطور سری بهم پیوسته‌اند، برای ایجاد ولتاژهای بالا (در حدود ۲۰۰-۴۰۰ ولت DC) تشکیل شده‌اند. وجود تفاوت‌هایی در خصوصیات شیمیایی سلول‌ها نظیر امپدانس داخلی و ظرفیت، عدم یکسانی دمای سلول‌های قرار گرفته درون یک استک و تفاوت‌های معمول در سیکل‌های شارژ و دشارژ، منجر به پدیده‌ی عدم تعادل شارژ بین سلول‌های مختلف یک استک باتری خواهد گشت. با توجه به اینکه سیستم کنترل شارژ باتری‌ها معمولاً با توجه به ولتاژ کل استک تصمیم‌گیری می‌کند، احتمال آنکه ولتاژ یک سلول مجزا در حین شارژ از حد بیشینه‌ی تعیین شده بالاتر رود و یا در حین دشارژ بیشتر از حد مجاز تخلیه گردد وجود دارد. باتری‌های لیتیوم-یون نمی‌توانند ولتاژی بالاتر از حدود ۴/۳ ولت را تحمل کنند. اگر یک سلول لیتیوم یون بیش از اندازه شارژ گردد، فشار و دمای آن سلول به شدت بالا رفته و می‌تواند حتی منجر به منفجر شدن سلول شود. درحالی‌که تخلیه بیش از حد سلول‌ها (کمتر از ۲/۴ ولت) باعث تغییراتی بازگشت‌ناپذیر در ساختار شیمیایی سلول شده و در نهایت موجب کاهش شدید ظرفیت مفید آن‌ها می‌گردد. در صورت وقوع چنین رخدادی، استک باتری تأمین‌کننده‌ی انرژی می‌تواند با پدیده‌هایی نظیر از دست دادن ظرفیت یک یا چند سلول و یا حتی انفجار روبرو گردد. بنابراین وجود یک سیستم مدیریت شارژ و دشارژ، که بتواند با پدیده‌ی عدم تعادل بین سلول‌های باتری مقابله کند بسیار حیاتی است.

از طرف دیگر از آنجا که پارامتر ولتاژ در حالتی که باتری تحت بار قرار دارد به تنهایی نمی‌تواند بیانگر سطح شارژ واقعی سلول باتری باشد، حتی اگر سیستم مدیریت باتری ولتاژ تک تک سلول‌ها را نیز در اختیار داشته باشد، باز نمی‌تواند تصمیمات درستی برای مدیریت شارژ سلول‌ها اتخاذ کند چرا که بسته به شرایط دمایی و میزان بار، مقدار ولتاژ باتری با سطح شارژ مورد انتظار متفاوت است. بنابراین استفاده از پارامتر ولتاژ در سیستم مدیریت باتری می‌تواند به عملکرد ضعیفی منجر شود و نیاز به بکارگیری روش‌هایی است

که با اندازه‌گیری جریان و ولتاژ پایانه باتری و استفاده از دینامیک مدل شده باتری، انرژی داخلی آن را در حالت کار بطور پیوسته تخمین بزنند. به این ترتیب با استفاده از این تخمین گر سطح شارژ و روش‌های کنترلی مناسب، می‌توان سطح شارژ باتری را به صورت مطلوب کنترل کرد.

در این فصل به بررسی مبانی تئوری و روش‌های مختلف مدل‌سازی و شناسایی پارامترهای باتری و تخمین سطح شارژ پرداخته شده و در نهایت روش‌های یکسان‌سازی سطح شارژ تشریح می‌گردد. به این منظور در ابتدا به مبانی تئوری باتری و انواع تکنولوژی‌های باتری پرداخته می‌شود. سپس ساختارهای مختلفی از مدل‌سازی باتری تشریح شده و روش‌های شناسایی پارامترهای یک باتری معرفی می‌گردد. پس از آن با بکارگیری دو روش ۱- تخمین با استفاده از فیلتر کالمن و ۲- تخمین با استفاده از ر‌ویت گر مد لغزشی، به تخمین سطح شارژ (SOC) باتری می‌پردازد. در ادامه روش‌های یکسان‌سازی سطح شارژ سلول‌های یک باتری تشریح می‌گردد و در نهایت ضمن جمع‌بندی مطالب فصل، روش مورد نظر در پروژه حاضر برای یکسان‌سازی سطح شارژ انتخاب می‌گردد.

۲-۱- مقدمه‌ای بر باتری

باتری الکتروشیمیایی به عنوان یکی از منابع ذخیره‌ساز انرژی قابل حمل، یک تجهیز مهم و پایه‌ای در خودروهای الکتریکی و الکتریکی-ترکیبی محسوب می‌شود. باتری وسیله‌ای است که انرژی شیمیایی موجود در مواد فعال خود را از طریق یک واکنش الکتروشیمیایی به انرژی الکتریکی تبدیل می‌کند و در مواقع نیاز به سیستم‌های الکتریکی تحویل می‌دهد. جزء پایه الکتروشیمیایی باتری، سلول نامیده می‌شود. یک باتری از دو یا چند سلول تشکیل شده است که از نظر الکتریکی به طور سری به هم متصل شده‌اند. هر سلول باتری شیمیایی از چهار المان اصلی تشکیل شده است: الکتروود مثبت، الکتروود منفی، صفحه جداکننده^۱ و الکترولیت. باتری را با ظرفیت کولومتری^۲ آن مشخص می‌کنند که عبارت است از مقدار آمپر-ساعتی که طی خالی شدن باتری از حالت شارژ کامل تا زمان رسیدن ولتاژ پایانه‌های آن به ولتاژ قطع می‌توان از باتری کشید [۳]. در ادامه تاریخچه‌ای از ساخت باتری و انواع مختلف باتری‌های مرسوم ارائه خواهد شد.

۲-۱-۱- تاریخچه باتری

قدمت استفاده از باتری به ۲۵۰ سال پیش از میلاد مسیح برمی‌گردد زمانی که در بغداد از ظروف مسی به عنوان سلول‌های گالوانیک استفاده می‌شد. این ظروف حاوی آهن و مس به عنوان الکتروود به همراه یک محلول اسید آلی بودند.

دو شخص که در توسعه علم الکتروشیمی و باتری نقش عمده‌ای داشته‌اند لوئیجی گالوانی^۳ و آلساندرو ولتا^۴ بوده‌اند [۳]. گالوانی در سال ۱۷۹۰ توانست توسط دو میله فلزی از جنس آهن و مس، عضلات پای یک قورباغه مرده را به حرکت درآورد. ولتا این پدیده را وابسته به ایجاد جریان الکتریکی بین الکتروودهای آهن و مس دانست. وی یک پیل طراحی کرد که شامل دو الکتروود نقره و روی بود و بین دو صفحه

¹ Separator

² Coulometric Capacity

³Luigi Galvani

⁴Alessandro Volta

الکترو، یک صفحه آغشته به محلول الکترولیت قرار داشت. او این اختراعش را در سال ۱۸۰۰ به ثبت رساند و در سال ۱۸۳۴ مایکل فاراده^۵ قوانین الکتروشیمی حاکم بر کار ولتا را به دست آورد. در حقیقت این کار فاراده، پلی را بین انرژی شیمیایی و الکتریکی برقرار ساخت.

جدیدترین مشکل در راه توسعه باتری، تشکیل گاز در الکترودها بود. مشکل از آنجا ناشی می‌شد که انرژی که صرف تشکیل گاز می‌گشت نمی‌توانست بازیافت شود. لکلانشه^۶ در سال ۱۸۶۶ بهترین راه را برای حل این مشکل یافت. او از اکسید منگنز که برای بهبود رسانایی با کربن مخلوط شده بود و بر روی یک صفحه گرافیتی قرار داشت، به عنوان الکترو مثبت استفاده نمود. وی همچنین از روی به عنوان الکترو منفی و از کلرید آلومینیوم به عنوان الکترولیت استفاده کرد. او توانست یک ولتاژ مدار باز ۱/۵ ولتی به کمک این باتری تولید نماید.

در سال ۱۸۹۵ تأثیر بکارگیری فلزات مختلف به عنوان الکترو و همچنین محلول با غلظت‌های مختلف اسید سولفوریک به عنوان الکترولیت توسط گاستون پلنت^۷ بررسی شد. او باتری را به شکل یک ساندویچ که از لایه‌های نازک هادی تشکیل شده بود ساخت. این صفحات فلزی توسط پارچه زبری جدا شده و در یک ظرف استوانه‌ای قرار داشتند که این ظرف با محلول اسید سولفوریک رقیق پر می‌شد و پس از شارژ و شکل‌گیری لایه‌های فعال می‌توانست ولتاژی در حدود ۲/۷ ولت ایجاد کند. این کار آغازی بر ایجاد باتری‌های قابل شارژ بود. یکی از اولین کاربردهای این گونه باتری‌ها برای روشنایی خانه‌ها بود که همراه با یک دینام که نقش شارژر را داشت توسط هنری تتودور در سال ۱۸۸۲ به کار گرفته شد.

بین سال‌های ۱۹۰۵-۱۸۹۵ والدمر جانگنر^۸ در سوئد و توماس ادیسون^۹ در آمریکا، باتری‌های نیکل-کادمیوم، آلکالاین و نیکل-آهن را پایه‌گذاری کردند. این دست از باتری‌ها نیز همانند باتری‌های سرب-اسید قابل شارژ بودند. آنچه که باعث استفاده از محلول آلکالاین به عنوان الکترولیت می‌شد، امکان استفاده از بازه وسیعی از مواد به عنوان الکترو بود.

تلاش‌های دیگری نیز در زمینه توسعه باتری صورت گرفته است. بیشترین تلاش‌ها برای بهبود پارامترهای مختلف باتری مانند چگالی انرژی، انرژی مخصوص، عمر مفید، نرخ تخلیه خودبخود و غیره بوده است. همچنین در زمینه توسعه باتری‌های قابل حمل نیز تلاش‌هایی صورت گرفته است. از جمله می‌توان به ایجاد باتری‌های نیکل-متال هیدرید در سال ۱۹۹۰ و باتری لیتیم-یون در سال ۱۹۹۱ اشاره کرد. دو عامل مهم دیگر که به توسعه و پیشرفت صنعت باتری انجامید، نیاز به چگالی انرژی بیشتر و همچنین توجه به مسائل زیست محیطی باتری است [۵].

۲-۱-۲- تکنولوژی‌های باتری

تاکنون تحقیقات زیادی بر روی باتری‌های شیمیایی صورت گرفته است. این باتری‌ها بر حسب ساختار، راندمان، طول عمر، سهولت استفاده و قیمت محصول تقسیم‌بندی می‌شوند. باتری‌های مورد استفاده در خودروهای الکتریکی و الکتریکی ترکیبی عبارت‌اند از: باتری‌های سرب-اسیدی^{۱۰}، باتری‌های

^۵Michael Faraday

^۶Leclanche

^۷Gaston Plante

^۸Waldmer Jungner

^۹Thomas Edison

^{۱۰}Lead-Acid

نیکلی اعم از نیکل-آهن^{۱۱}، نیکل-کادمیوم^{۱۲} و نیکل-فلز و باتری‌های لیتیومی مانند لیتیوم-پلیمر (Li-Pol)^{۱۳} و لیتیوم-یون (Li-ion)^{۱۴} [۶].

باتری‌های سرب-اسید

این باتری‌ها یکی از متداولترین باتری‌های مورد استفاده در خودروهای الکتریکی ترکیبی هستند. یک سلول سرب-اسیدی دارای پنج جزء اصلی است که عبارتند از الکترود منفی، الکترود مثبت، الکترولیت، صفحه جداکننده و جداره‌های سلول. الکترود منفی از جنس سرب اسفنجی و الکترود مثبت در هنگام شارژ کامل، از جنس دی‌اکسید سرب (PbO_2) می‌باشد. در طول زمان تخلیه شدن، الکترودها از سمت بار به سمت الکترود مثبت جاری می‌شوند و این موضوع منجر به تغییر ترکیب شیمیایی این الکترود به سولفات سرب ($PbSO_4$) می‌شود. الکترولیت، مدار داخل یک باتری را با تأمین کردن یون‌ها برای الکترودهای مثبت و منفی تکمیل می‌کند. اسید سولفوریک رقیق (H_2SO_4) در باتری‌های سرب-اسیدی به عنوان الکترولیت به کار می‌رود. در حالت شارژ کامل این محلول تقریباً متشکل از ۲۵٪ اسید و ۷۵٪ آب است. یک صفحه جداکننده در هر سلول باتری استفاده می‌شود تا الکترودهای مثبت و منفی را از نظر الکتریکی از یکدیگر جدا کند. جنس این صفحه باید به گونه‌ای باشد که اجازه انتقال یون بین الکترودها و الکترولیت را بدهد [۳]. به طور کلی بسته به روش ساخت باتری، دو نوع باتری سرب-اسیدی وجود دارد که عبارتند از: باتری شناور و باتری سربسته. این دو نوع باتری از نظر عملکرد نیز متفاوت هستند.

تمامی باتری‌های سرب-اسیدی بر اساس یک واکنش شیمیایی یکسان عمل می‌کنند. زمانی که باتری تخلیه می‌شود، مواد فعال در الکترودها (دی‌اکسید سرب در الکترود مثبت و سرب اسفنجی در الکترود منفی) با اسید سولفوریک الکترولیت واکنش می‌دهد و منجر به تولید سولفات و آب می‌شوند. در زمان شارژ سولفات سرب در هر الکترود دوباره به دی‌اکسید سرب (الکترود مثبت) و سرب اسفنجی (الکترود منفی) تبدیل می‌شود و یون‌های سولفات (SO_4^{2-}) دوباره به محلول الکترولیت برمی‌گردند تا اسید سولفوریک شکل گیرد [۳].

باتری‌های نیکلی

نیکل نسبت به سرب سبک‌تر است و از خواص الکتروشیمیایی بسیار خوبی برخوردار می‌باشد. این نوع باتری‌ها به چهار گروه اصلی تقسیم می‌شوند: نیکل-آهن، نیکل-روی، نیکل-کادمیوم و نیکل-هیدرید فلز [۶].

باتری نیکل-آهن متشکل از الکترود مثبتی از جنس نیکل-هیدروکسی‌اکسید ($NiOOH$)، الکترود منفی آن‌ها از جنس آهن نقره‌ای و الکترولیت آن محلول غلیظ هیدروکسید پتاسیم (معمولاً ۲۴۰ gr/lit) حاوی هیدروکسید لیتیوم (۵۰ gr/lit) می‌باشد. از مزایای این باتری می‌توان به چگالی توان بالا نسبت به باتری‌های سرب-اسیدی و قابلیت تحمل عمیق آن اشاره کرد.

باتری نیکل-کادمیوم انرژی مخصوصی در همان حدود باتری سرب-اسیدی دارد. الکترود مثبت و الکترولیت در این نوع باتری‌ها مثل باتری‌های نیکل آهن می‌باشد. الکترود منفی از کادمیوم فلزی تشکیل

¹¹Nickel-Iron

¹² Nickel-Metal

¹³Lithium-Polymer

¹⁴Lithium-Ion

شده است. از مزایای این باتری‌ها می‌توان به توان مخصوص بالا (بیشتر از 220 W/kg)، چرخه عمر طولانی (بالتر از ۲۰۰۰ سیکل)، تولرانس بالای مکانیکی و الکتریکی، افت ولتاژ کم در گستره وسیعی از جریان‌های تخلیه، قابلیت شارژ سریع، گستره دمای کاری وسیع، نرخ تخلیه خودبخودی^{۱۵} بسیار پایین، خوردگی ناچیز و وجود در اندازه‌های مختلف اشاره کرد.

باتری‌های نیکل-هیدرید فلز مشخصه‌ها و خصوصیات شبیه به باتری‌های نیکل-کادمیوم دارند، اما تفاوت عمده بین این دو باتری این است که در این نوع از باتری‌ها از هیدروژن موجود در هیدرید فلز به عنوان ماده فعال به جای کادمیوم استفاده می‌شود. باتری‌های نیکل-هیدرید فلز دارای انرژی مخصوص و ظرفیتی بیشتر از باتری‌های نیکل-کادمیوم بوده و کارکرد ولتاژی آنها مثل باتری‌های نیکل-کادمیوم می‌باشند.

باتری‌های لیتیومی

لیتیم سبک‌ترین فلز می‌باشد و از نظر مشخصه‌های الکتروشیمیایی بسیار عالی می‌باشد. ولتاژ ترمودینامیکی بسیار بالا و در نتیجه انرژی و توان مخصوص بسیار بالایی دارد. این نوع باتری‌ها از دو دسته عمده تشکیل می‌شوند: لیتیم-پلیمر و لیتیم-یون [۶].

در باتری لیتیم-پلیمر از نوعی ماده که قابلیت هدایت یون را دارد به عنوان الکترولیت استفاده شده است. این باتری دارای انرژی مخصوص (انرژی در واحد جرم) و توان مخصوص بالایی است. الکتروند منفی از فلز لیتیم درست شده است و الکتروند مثبت از اکسید فلز واسطه $(M_y O_z)$ یک ساختار لایه‌ای دارد که می‌توان یون‌های لیتیم را در آن جای داد (حین فرایند تخلیه شدن) و یا از آن خارج نمود (حین فرایند شارژ شدن). در هنگام تخلیه یون‌های لیتیم تشکیل شده در الکتروند منفی از طریق الکترولیت جامد به الکتروند مثبت می‌روند و در ساختار کریستالی آن جای می‌گیرند. در هنگام شارژ شدن، این عملیات به طور معکوس انجام می‌گیرد. یکی از این باتری‌ها که بیشترین استفاده را دارد باتری $Li / SPE / V_6O_{13}$ است. مشخصه‌های اصلی این باتری‌ها عبارتند از: انرژی مخصوصی معادل 155 Wh/kg و توان مخصوصی برابر 315 W/kg .

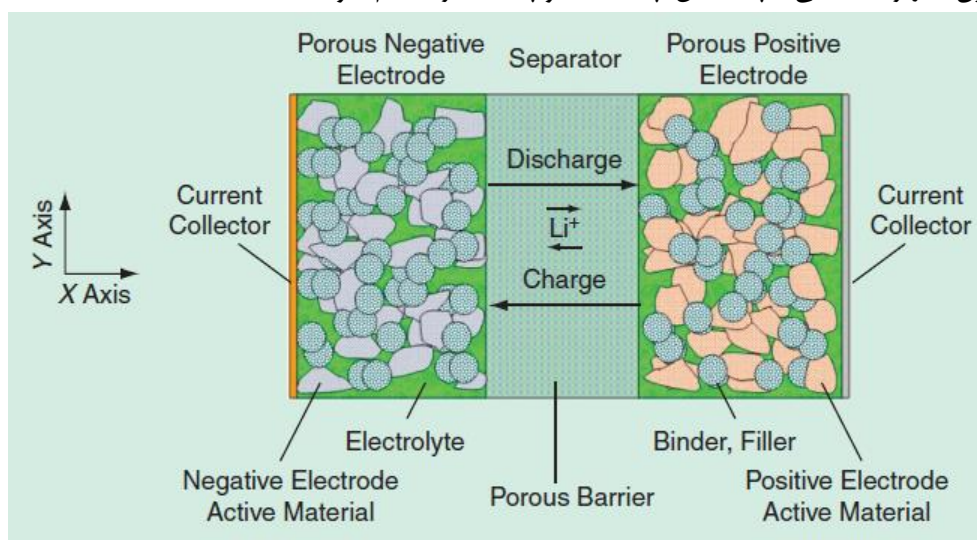
باتری لیتیم-یون برای خودروهای الکتریکی و الکتریکی ترکیبی بسیار مناسب است؛ چرا که دارای انرژی مخصوص بالا، توان مخصوص بالا، بازده انرژی بالا، بازده بالا در دماهای زیاد است و قابل بازیافت نیز می‌باشد [۲-۳]. یک باتری لیتیم-یون دارای چهار جزء اصلی است (شکل ۱-۱). الکتروند مثبت این نوع از باتری‌ها از موادی با رابطه شیمیایی کلی $Li_{1-x} M_y O_z$ و الکتروند منفی از کربن-لیتیم $Li_x C$ یا معمولاً از گرافیت C (که یک ماده بین‌لایه‌ای^{۱۶} است) تشکیل شده است. الکتروند مثبت به پایانه مثبت و الکتروند منفی به پایانه منفی سلول متصل شده است. یک صفحه جداکننده که یک واسطه مشبک نازک می‌باشد، الکتروند مثبت را از الکتروند منفی جدا می‌کند. این صفحه جداکننده که یک عایق الکتریکی است، اجازه نمی‌دهد الکترون‌ها بین الکتروند مثبت و منفی جریان پیدا کنند، اما به دلیل مشبک بودن، اجازه عبور یون‌ها را به وسیله الکترولیت می‌دهد [۲]. الکترولیت نوعی مایع حلال آلی است که از ذرات باردار (یون Li^+) تشکیل شده است. این ذرات باردار می‌توانند در پاسخ به یک تغییر پتانسیل الکتروشیمیایی حرکت

¹⁵ Self Discharging

¹⁶ Intercalation material

کنند. الکترولیت‌های مایع در باتری‌های لیتیم-یون از نمک‌های لیتیم مانند LiPF_6 ، LiBF_4 یا LiClO_4 در یک حلال آلی مانند اتیلن کربنات، دی‌متیل کربنات و دی‌اتیل کربنات تشکیل شده است. در برخی باتری‌های Li-ion ، الکترولیت یک جامد پلیمری است که هم به عنوان واسط هادی یون و هم به عنوان جداکننده عایق الکتریکی به کار می‌رود. الکترولیت مثبت، الکترولیت منفی و جداکننده همگی در داخل الکترولیت فرو رفته‌اند و الکترولیت تمام منافذ مواد جامد را پر می‌کند. در هر دو نوع این باتری‌ها، ذرات باردار داخل باتری یون Li^+ است. در هنگام تخلیه، یون‌های لیتیم از الکترولیت منفی (گرافیت) جدا شده و به الکترولیت مثبت وارد می‌شوند. در هنگام شارژ، این عمل به صورت برعکس اتفاق می‌افتد.

از مواد به کار رفته در الکترولیت مثبت می‌توان به موارد $\text{Li}_{1-x}\text{CoO}_2$ ، $\text{Li}_{1-x}\text{NiO}_2$ و $\text{Li}_{1-x}\text{Mn}_2\text{O}_4$ اشاره کرد که دارای مزایایی از قبیل پایداری در هوا، ولتاژ بالا و قابلیت بازگشت‌پذیری در واکنش‌ها هستند. بر اساس این که در الکترولیت مثبت کدام یک از موارد بالا به کار رفته باشد، باتری لیتیم-یون به زیرشاخه‌هایی با پایه نیکل، پایه کبالت و پایه منگنز تقسیم نمود.



شکل ۱-۱- شماتیک یک سلول لیتیم-یون [۲]

در میان تکنولوژی‌های معرفی شده برای باتری، باتری‌های لیتیم-یون به دلیل عملکرد خوب، چگالی انرژی بالا و قابلیت اعتماد بالایشان به طور گسترده در قطعات الکترونیکی قابل حمل زیادی از جمله موبایل، دوربین‌های دیجیتال و لپ‌تاپ‌ها مورد استفاده قرار می‌گیرند [۷]. در مقایسه با تکنولوژی‌های دیگر باتری، باتری‌های لیتیم-یون بهترین نسبت انرژی به وزن را فراهم کرده و دارای دشارژ خودبخودی کمی می‌باشند [۲]. در مقایسه با باتری‌های سرب-اسیدی، SOC آن‌ها تغییرات بیشتری می‌تواند داشته باشد، به طوری که در باتری سرب-اسیدی حداکثر SOC بین ۴۰٪ تا ۹۰٪ است، ولی در باتری‌های لیتیم-یون محدوده SOC حدود ۱۰٪ تا ۱۰۰٪ می‌باشد [۱]. این خواص سودمند به همراه هزینه‌های رو به کاهش، باتری‌های لیتیم-یون را به عنوان یک انتخاب عمده برای نسل‌های بعدی صنایع خودرو و صنایع فضایی قرار داده است. در بخش خودرو، تقاضای رو به افزایش برای HEV و EV، سازندگان خودرو را مجبور کرده است تا تکنولوژی‌های قابل اعتماد دیگری مانند باتری‌های لیتیم-یون را مد نظر قرار دهند تا آن را جایگزین تکنولوژی‌های معاصر باتری مانند باتری‌های سرب-اسید و باتری‌های نیکل-هیدرید فلز کنند. از آنجایی که باتری‌های لیتیم-یون امکان احتراق یا انفجار را در شرایط شارژ اضافی یا سوء مصرف دارند، طراحی BMS مناسب برای این باتری‌ها نیز مسئله بسیار مهمی است [۲].

۳-۲-۱- پارامترهای باتری

برخی از پارامترهای مهم یک باتری عبارتند از: ظرفیت باتری، سطح شارژ باتری، سطح سلامت باتری و دشارژ خودبخود باتری [۳].

ظرفیت باتری

ظرفیت باتری مقدار بار آزاد تولید شده در داخل باتری است که واحد آن آمپر ساعت (Ah) می باشد و هر آمپر ساعت معادل ۳۶۰۰ کولن در نظر گرفته می شود. ظرفیت باتری را می توان به صورت زیر بیان نمود:

$$Q_T = x \times n \times F \quad (1-1)$$

که در آن x تعداد مول تحت واکنش در کل طول فرایند تخلیه شدن، n تعداد الکترون های تولید شده توسط الکتروود منفی و $F = 96412 \frac{C}{mol}$ ثابت فارادیه می باشد. این رابطه را به صورت زیر هم می توان نوشت:

$$Q_T = 0.278F \frac{m_R n}{M_m} \quad (1-2)$$

که در آن m_R جرم مواد تحت واکنش بر حسب کیلوگرم و M_m جرم مولی مواد تحت واکنش می باشند.

در عمل به خاطر محدودیت های عملی مانند نحوه شارژ باتری، ظرفیت عملی Q_p در مقایسه با ظرفیت تئوری Q_T همیشه از مقدار کمتری برخوردار است. ظرفیت عملی باتری از رابطه زیر به دست می آید:

$$Q_p = \int_{t_0}^{t_{cut}} i(t) dt \quad (1-3)$$

که در آن t_0 لحظه ای است که باتری به طور کامل شارژ است و t_{cut} لحظه ای است که ولتاژ پایانه باتری به مقدار انتهایی خود (V_{cut}) می رسد و $i(t)$ جریان باتری است.

سطح شارژ باتری

سطح شارژ عبارت است از نسبت ظرفیت باقیمانده در یک باتری به ظرفیت کل آن. البته ظرفیت باتری شدیداً به مقدار جریان کشیده شده از آن وابسته است. به عنوان مثال اگر جریان کشیده شده از یک باتری کاهش یابد، ظرفیت آن نیز افزایش می یابد. لذا می توان تغییرات SOC (بر حسب Ah) را به صورت زیر تعریف کرد:

$$SOC_T(t) = Q_T - \int_0^t i(\tau) d\tau \quad (1-4)$$

با تقسیم طرفین بر ظرفیت عملی باتری Q_p داریم:

$$SOC(t) = SOC_0 - \int_0^t \frac{i dt}{Q_p} \quad (1-5)$$

که SOC_0 وضعیت اولیه شارژ در باتری است.

سطح سلامت باتری

وضعیت سلامت باتری (SOH) عبارت است از توانایی باتری برای ذخیره انرژی، دریافت و تحویل جریان‌های بالا و نگهداری شارژ در زمان‌های طولانی [۸]. کل شارژ ذخیره شده در یک سلول کاملاً شارژ شده، با استفاده مکرر از باتری کاهش می‌یابد، زیرا مواد فعال موجود در صفحات سلول با مکانیزم‌هایی همچون کاهش سطح فعال سلول، کم می‌شود. این کاهش ظرفیت با کاهش SOH نشان داده می‌شود. پارامتر SOH می‌تواند از روی ظرفیت، مقاومت داخلی و دیگر پارامترهای باتری مانند امپدانس AC، نرخ دشارژ خودبخودی و چگالی توان به دست آید. SOH از روی ظرفیت به صورت زیر تعریف می‌شود:

$$SOH(t) = \frac{C_{act}}{C_{nom}} \times 100\% \quad (1-6)$$

که C_{act} ظرفیت واقعی باتری و C_{nom} ظرفیت نامی آن است.

دشارژ خودبخود باتری

دشارژ خودبخودی پدیده‌ای در باتری است که در آن واکنش‌های شیمیایی داخلی، شارژ ذخیره شده در باتری را بدون هیچ اتصالی بین الکترودها کاهش می‌دهد. سرعت این پدیده در باتری به نوع باتری، سطح شارژ، جریان شارژ، دمای محیط و عوامل دیگر بستگی دارد. در میان باتری‌های قابل شارژ، باتری‌های لیتیم-یون کمترین نرخ دشارژ خودبخودی (حدود ۲ تا ۳٪ دشارژ در ماه) را دارند، در حالی که برخی باتری‌های بر پایه نیکل شدیداً تحت تأثیر این پدیده هستند (حدود ۳۰٪ برای باتری نیکل-هیدرید فلز).

۳-۱- مدل‌سازی باتری و شناسایی پارامترهای باتری

برای طراحی بهتر یک سیستم مدیریت باتری، به یک مدل دقیق برای توصیف مشخصه‌های مورد نیاز باتری نیاز است. از طرفی یکی از نگرانی‌های اصلی در طراحی سیستم‌های الکترونیکی قابل حمل، چگونگی کاهش اتلاف توان و افزایش زمان اجرای^{۱۷} باتری می‌باشد. بدون در دست داشتن مدار و مدل‌های باتری، پیش‌بینی و بهینه‌سازی زمان اجرای باتری و عملکرد مدار ممکن نخواهد بود [۱]. ساختارهای متعددی از مدل‌های باتری با درجات مختلفی از پیچیدگی برای اهداف گوناگون در دسترس است. این مدل‌ها را می‌توان به چهار دسته مدل‌های تجربی، مدل‌های الکتروشیمیایی، مدل‌های مدار الکتریکی و مدل‌های انتزاعی دسته‌بندی کرد [۷]. در سیستم مدیریت باتری عموماً از مدل مدار معادل در محاسبات و تخمین پارامترها استفاده می‌شود؛ لذا در اینجا بر روی مدل‌های الکتریکی باتری تمرکز شده است.

مدل‌های الکتروشیمیایی عمده‌تاً برای بهینه‌سازی جنبه‌های طراحی فیزیکی باتری، مشخص کردن مکانیزم اصلی تولید توان و ارتباط بین پارامترهای طراحی با خواص ماکروسکوپی (مانند ولتاژ و جریان باتری) و میکروسکوپی (مانند توزیع غلظت) به کار می‌روند. مدل‌سازی الکتروشیمیایی نه تنها قادر به پیش‌بینی کمیت‌های ماکروسکوپی مانند ولتاژ و جریان سلول است، بلکه می‌تواند توزیع محلی غلظت، پتانسیل، جریان و دمای داخل سلول را در یک مقیاس میکروسکوپی پیش‌بینی کند. این مدل‌ها عموماً دارای تعداد پارامترهای زیادی هستند که باید با استفاده از آزمایش‌هایی بر روی ترکیبات مختلف سلول

¹⁷ Runtime

تعیین شوند [۳ و ۹]. همچنین این مدل ها پیچیده و زمان بر هستند، چون شامل یک سیستم معادلات دیفرانسیل جزئی فضایی متغیر با زمان کوپل شده^{۱۸} هستند که برای حل آن ها نیاز به روزها شبیه سازی، الگوریتم های پیچیده عددی و اطلاعات خاص از باتری است که دستیابی به آن ها مشکل است [۱، ۷ و ۱۰].

در مدل های تجربی رفتار باتری توسط یکسری معادلات و پارامترها توصیف می شود. این پارامترها طوری محاسبه می شوند که داده های مدل بر داده های تجربی منطبق شوند. این مدل ها از رویکردهای آماری و معادلات تجربی استفاده می کنند و می تواند زمان اجرا، بازده و ظرفیت را پیش بینی کند، اما قادر به پیش بینی مشخصه I-V که برای شبیه سازی و بهینه سازی مدار مهم است، نمی باشد. این مدل ها بسیار مختصر و ساده شده هستند، ولی همچنان برای طراحان سیستم قابل استفاده می باشند. اغلب این مدل ها فقط در موارد خاصی مانند ارزیابی ظرفیت های غیردقیق کاربرد دارند و نتایج آن ها دارای دقت کمی است (خطای حدود ۵٪-۲۰٪) [۱ و ۱۰].

برخلاف مدل های دیگر، مدل های انتزاعی بر روش های صرفاً ریاضی برای توصیف رفتار باتری مبتنی می باشد. مدل های انتزاعی مبتنی بر هوش مصنوعی (AI)^{۱۹} شامل مدلسازی شبکه عصبی مصنوعی به همراه SVM^{۲۰} می باشد که می تواند بسیار دقیق باشد اما وابستگی زیادی به داده یادگیری دارد و ممکن است بدون در دست داشتن داده های یادگیری اضافی قابل توسعه به باتری های دیگر نباشد [۷]. اغلب مدل های انتزاعی ممکن به صورت عملی بی معنی باشند و فقط در کاربردهای خاصی کار کنند، اما هنوز برای مهندسين برق مفید هستند [۷].

۱-۳-۱- مدل های الکتریکی باتری

مدل های الکتریکی دارای دقتی کمتر از مدل های الکتروشیمیایی و بیشتر از مدل های تجربی هستند (خطای حدود ۱٪-۵٪) [۱]. این مدل ها، مدل های مدار معادل الکتریکی هستند که برای طراحی و شبیه سازی همراه با سایر مدارها و سیستم های الکتریکی به کار می روند و از ترکیب منابع ولتاژ، مقاومت ها و خازن ها برای تقریب فرایندهای الکتروشیمیایی و دینامیک های یک باتری استفاده می کنند. برای یک مهندس برق، مدل های الکتریکی محسوس تر، مفیدتر و دارای استفاده آسان است. مدل های الکتریکی زیادی از باتری ها، از باتری سرب-اسید تا لیتیوم-یون وجود دارد. بیشتر این مدل ها در سه دسته اساسی مدل های بر مبنای تونن، بر مبنای امپدانس و بر مبنای زمان اجرا^{۲۱} قرار می گیرند [۱۱ و ۱۴].

مدل های الکتریکی بر مبنای تونن

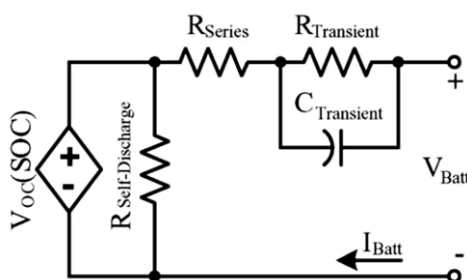
مدل بر مبنای تونن، در ساده ترین فرم (شکل ۱-۲)، از یک مقاومت سری (R_{Series}) و یک شبکه RC موازی ($R_{Transient}$ و $C_{Transient}$) برای پیش بینی پاسخ باتری به بارهای گذرا در یک سطح شارژ مشخص و با در نظر گرفتن ولتاژ مدار باز ثابت (V_{OC} (SOC)) استفاده می کند. این فرض باعث می شود نتوان تغییرات ولتاژ حالت ماندگار باتری (پاسخ dc) و نیز اطلاعات زمان اجرا را به دست آورد [۱].

¹⁸ Coupled time-variant spatial partial differential equations (PDEs)

¹⁹ Artificial Intelligence

²⁰ Support Vector Machine

²¹ Runtime

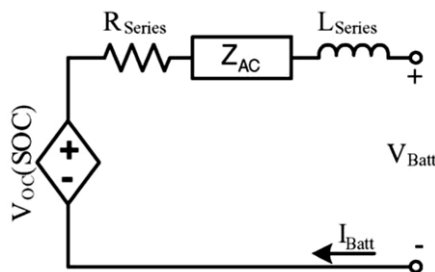


شکل ۱-۲- مدل الکتریکی بر مبنای تونن [۱]

در مدل‌های اصلاح شده با اضافه کردن عناصر اضافی برای پیش‌بینی زمان اجرا و پاسخ dc بهبودهایی حاصل شده است. هر چه تعداد شبکه‌های RC موازی بیشتر باشد دقت پاسخ پیش‌بینی شده بیشتر خواهد بود. در [۱۸] از یک خازن متغیر به جای $V_{OC}(SOC)$ برای نشان دادن ولتاژ مدار باز غیرخطی و SOC استفاده شده است که پارامتر خازن را پیچیده می‌کند. برای به دست آوردن SOC نیاز به انتگرال‌گیری روی ولتاژ دارد و تقریباً ۵٪ خطای زمان اجرا و ۰/۴ ولت خطای ولتاژ را به ازای جریان‌های شارژ و دشارژ ثابت را خواهیم داشت. [۱۹] رابطه غیرخطی بین ولتاژ مدار باز و SOC را مدل می‌کند، ولی رفتار گذرا را نادیده می‌گیرد؛ در [۲۰-۲۲] با معادلات ریاضی اضافی می‌توان SOC و تخمین زمان اجرا را به دست آورد و این معادلات در شبیه‌سازی‌های مدار مدل نشده‌اند. در [۲۳] از دو شبکه RC موازی ثابت استفاده شده است، ولی فقط برای شرایط دمایی و SOC مشخص کار می‌کند. [۲۴] از یک شبکه الکتریکی پیچیده برگرفته از فرایند فیزیکی برای مدل‌سازی ولتاژ مدار باز استفاده می‌کند.

مدل‌های الکتریکی بر مبنای امپدانس

مدل‌های بر مبنای امپدانس (شکل ۱-۳) از روش طیف‌نمایی امپدانس الکتروشیمیایی برای به دست آوردن یک مدل امپدانس معادل ac در حوزه فرکانس استفاده کرده و سپس با استفاده از یک شبکه معادل پیچیده (Z_{ac})، طیف امپدانس را متناسب می‌کند. فرایند منطبق کردن، سخت، پیچیده و نامحسوس است. علاوه بر این، مدل‌های بر مبنای امپدانس همانند مدل‌های تونن، فقط برای SOC و تنظیمات دمایی مشخصی دقیق هستند؛ بنابراین دقت آن‌ها هنگام پیش‌بینی پاسخ dc و زمان اجرای باتری محدود می‌شود [۱ و ۱۴].

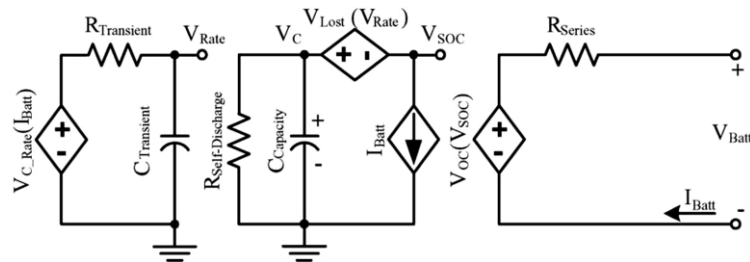


شکل ۱-۳- مدل الکتریکی بر مبنای امپدانس [۱]

مدل‌های الکتریکی بر مبنای زمان اجرا

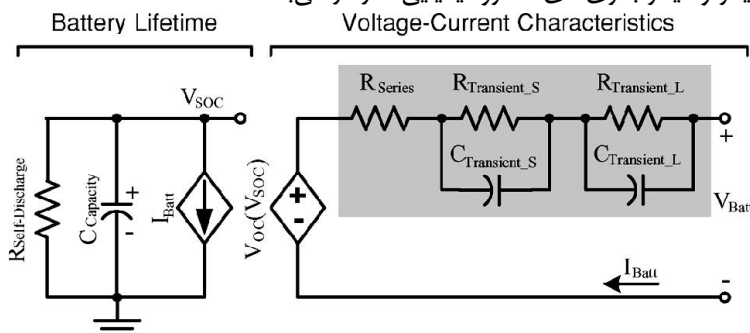
مدل‌های الکتریکی بر مبنای زمان اجرا (شکل ۱-۴) به دو صورت زمان پیوسته یا زمان گسسته پیاده‌سازی می‌شود تا پاسخ زمان اجرا و ولتاژ dc باتری را برای جریان‌های دشارژ ثابت در شبیه‌سازهای

SPICE شبیه‌سازی کند. هر چه جریان بار تغییر کند عدم دقت افزایش می‌یابد [۱۰]. برای پیاده‌سازی مدل زمان پیوسته از شبیه‌سازهای SPICE و برای زمان گسسته از کد VHDL^{۲۲} استفاده می‌شود [۱].



شکل ۴-۱- مدل الکتریکی بر مبنای زمان اجرا [۱]

در [۱] یک مدل الکتریکی دقیق، محسوس و جامع برای تخمین زمان اجرا و مشخصه $I-V$ باتری ارائه شده است (شکل ۵-۱). در این مدل تمام مشخصه‌های دینامیکی باتری از ولتاژ مدار باز غیرخطی، جریان، دما، تعداد سیکل و ظرفیت ذخیره وابسته به زمان تا پاسخ گذرا قابل دستیابی هستند. این مدل تمام خواص مهم باتری را پیش‌بینی می‌کند و با باتری‌های سرب-اسید، نیکل-کادمیوم، نیکل-هیدرید فلز، لیتیم-یون، لیتیم-پلیمر و دیگر باتری‌های الکتروشیمیایی سازگار می‌باشد.



شکل ۵-۱- یک مدل الکتریکی دقیق بر مبنای زمان اجرا از باتری [۱]

در سمت چپ خازن $C_{Capacity}$ و منبع جریان کنترل شده با جریان برگرفته از مدل‌های بر مبنای زمان اجرا، ظرفیت، SOC و زمان اجرای باتری را مدل می‌کند. شبکه RC، مشابه مدل‌های بر مبنای تونن، پاسخ گذرا را شبیه‌سازی می‌کند. برای بیان ارتباط بین SOC و ولتاژ مدار باز، از یک منبع ولتاژ کنترل شده با ولتاژ استفاده شده است. این مدل، ترکیبی از مدل‌های الکتریکی قبلی باتری می‌باشد. این امر باعث می‌شود تا بتوان زمان اجرا، پاسخ حالت ماندگار و پاسخ گذرای باتری را به طور دقیق پیش‌بینی کرد و همه مشخصات دینامیکی باتری مثل ظرفیت قابل استفاده ($C_{Capacity}$)، ولتاژ مدار باز (V_{OC}) و پاسخ گذرای (شبکه RC) را به دست آورد.

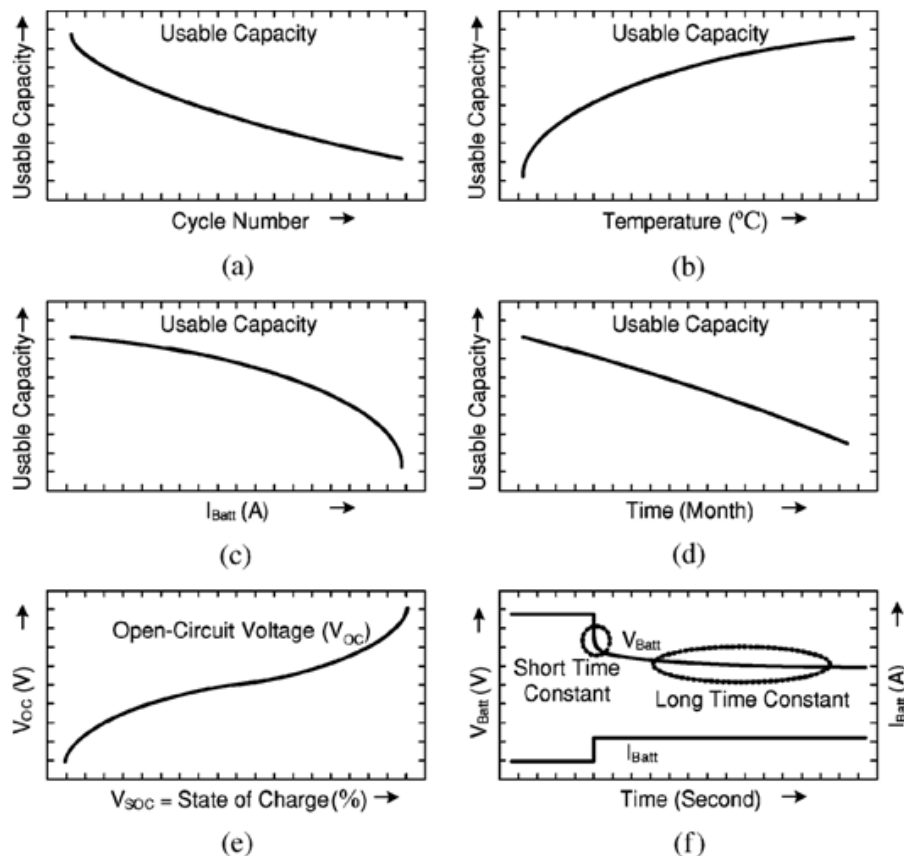
با این فرض که باتری از حالت شارژ اولیه یکسان تا ولتاژ پایان دشارژ مشابهی تخلیه شود، انرژی گرفته شده که همان ظرفیت قابل استفاده باتری است با افزایش تعداد سیکل‌ها، جریان دشارژ و زمان ذخیره (به دلیل دشارژ خودبخود) و نیز کاهش دما، کاهش می‌یابد (شکل ۶-۱ (a-d)). ظرفیت قابل استفاده را می‌توان با یک خازن ظرفیت کامل ($C_{Capacity}$)، یک مقاومت دشارژ خودبخود (

²²Very high speed integrated circuit Hardware Description Language

کرد. $C_{Capacity}$ نشان‌دهنده کل شارژ ذخیره شده در باتری یعنی SOC است و به صورت زیر تعیین می‌شود:

$$C_{Capacity} = 3600 \cdot Capacity \cdot f_1(Cycle) \cdot f_2(Temp) \quad (1-7)$$

که در آن $Capacity$ ظرفیت نامی باتری بر حسب آمپر-ساعت است و $f_1(Cycle)$ و $f_2(Temp)$ ضرایب تصحیح وابستگی به تعداد سیکل و دما است (شکل ۱-۶ (a و b)). تغییر ظرفیت قابل استفاده بر حسب جریان به دلیل مقادیر SOC مختلف در پایان دشارژ به ازای جریان‌های مختلف است (شکل ۱-۶ (c)). هنگامی که باتری شارژ یا دشارژ می‌شود، منبع جریان وابسته به جریان I_{Batt} ، خازن $C_{Capacity}$ را شارژ یا دشارژ می‌کند و SOC که با V_{SOC} نشان داده شده است، به طور دینامیکی تغییر می‌کند. بنابراین هنگامی که ولتاژ باتری به ولتاژ پایان دشارژ رسید، زمان اجرای باتری به دست می‌آید.



شکل ۱-۶- منحنی‌های مشخصه باتری؛ ظرفیت قابل استفاده باتری بر حسب (a) تعداد سیکل، (b) دما، (c) جریان، (d) زمان ذخیره‌سازی و (e) ولتاژ مدار باز بر حسب SOC و پاسخ گذرای باتری به جریان بار پله‌ای [۱]

مقاومت دشارژ خودبخود ($R_{Self_Discharge}$) برای توصیف اتلاف انرژی به دلیل دشارژ خودبخود است هنگامی که باتری برای زمان‌های طولانی به کار گرفته می‌شود. از نظر تئوری $R_{Self_Discharge}$ تابعی از SOC ، دما و اغلب تعداد سیکل است، اما در عمل می‌تواند به صورت یک مقاومت بزرگ در نظر گرفته

شود و یا حتی صرف نظر شود، چون مطابق شکل ۶-۱ (d) ظرفیت قابل استفاده به طور آهسته با زمان کاهش می‌یابد.

ولتاژ مدار باز V_{OC} با تغییر SOC به صورت غیرخطی تغییر می‌کند (شکل ۶-۱ (e)). این رابطه غیرخطی با منبع ولتاژ وابسته به ولتاژ $V_{OC}(V_{SOC})$ مدل می‌شود. ولتاژ مدار باز معمولاً به عنوان ولتاژ حالت ماندگار در نقاط مختلف SOC اندازه‌گیری می‌شود.

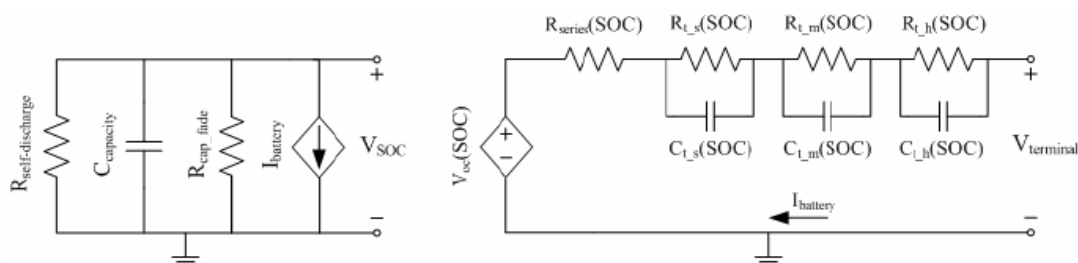
در جریان بار پله‌ای، ولتاژ باتری پاسخ کندی را از خود نشان می‌دهد (شکل ۶-۱ (f)). برای توصیف پاسخ گذرای باتری از شبکه RC در شکل ۵-۱ استفاده شده است. R_{Series} دلیل افت ولتاژ ناگهانی در پاسخ پله می‌باشد. همچنین ثابت زمانی‌های کوتاه و طولانی موجود در پاسخ پله به دلیل وجود $R_{Transient_S}$ ، $C_{Transient_S}$ ، $R_{Transient_L}$ و $C_{Transient_L}$ می‌باشد.

تمام پارامترهای مدل ارائه شده تابعی چندمتغیره از SOC، دما، جریان و تعداد سیکل هستند و در نتیجه استخراج مدل پیچیده می‌شود. برای ساده‌سازی مدل، برخی از این پارامترها در باتری‌های لیتیم-یون قابل صرف نظر هستند؛ مثلاً وابستگی ظرفیت قابل استفاده باتری به دشارژ خودبخود (۲٪-۱۰٪ در ماه) و تعداد سیکل (کمتر از ۱۰٪ اتلاف ظرفیت در طی ۳۰۰ سیکل). بنابراین $R_{Self_Discharge}$ را برابر بینهایت و $f_1(Cycle)$ را برابر یک در نظر می‌گیریم. همچنین با ثابت نگه داشتن دمای باتری تمام پارامترها مستقل از دما خواهند بود و $f_2(Temp)$ می‌تواند برابر ۱ در نظر گرفته شود.

برای ارزیابی مدل استخراج شده از باتری، سه پروفایل بار مختلف (دشارژ پیوسته، شارژ پالسی و دشارژ تناوبی چهار پله‌ای) به باتری پلیمر لیتیم-یون اعمال شده است. نزدیکی قابل قبول بین نتایج شبیه‌سازی و داده‌های تجربی نشان می‌دهد که این مدل به طور دقیق زمان اجرای باتری را با خطای کمتر از ۴٪ و پاسخ ولتاژ را با خطای کمتر از ۳۰ میلی‌ولت برای هر پروفایل باری پیش‌بینی می‌کند.

در [۱۰] یک مدل مبتنی بر زمان اجرای باتری که قادر به بازسازی مشخصه I-V آن می‌باشد ارائه شده است. این مدل دارای چند ثابت زمانی در گستره‌های ثانیه، دقیقه و ساعت می‌باشد؛ علاوه بر این ثابت‌های زمانی، یک ثابت زمانی در گستره میلی‌ثانیه نیز محقق می‌شود.

برای تعیین دقیق ظرفیت دشارژ که تابعی از شدت دشارژ $i(t)$ ، دما $f_2[T(t)]$ و تعداد سیکل $f_3[n_{cycle}]$ می‌باشد، ضریب شدت $f_1[i(t)]$ نیز لحاظ شده است. ضریب شدت، میزان کاهش ظرفیت را در اثر واکنش‌های جانبی^{۲۳} ناخواسته هنگام افزایش جریان به حساب می‌آورد. این مدل در شکل ۷-۱ نشان داده شده است.



شکل ۷-۱- مدل الکتریکی باتری برای استفاده در شبیه‌سازی HEV در یک دمای ثابت [۱۰]

²³Side Reaction

معادلات حاکم بر این مدل به صورت زیر است:

$$SOC[i(t), T(t), n_{cycle}, t] = SOC_{initial} + \int_0^t f_1[i(t)] \times f_2[T(t)] \times f_3[n_{cycle}] \times i(t) dt \quad (1-8)$$

$$V_{terminal} = V_{oc}(SOC, T) - i_{bat}(t) \times R_{in}(SOC, T) + i_{bat}(t) \times R_{transient}(\tau_s, \tau_m, \tau_h) \quad (1-9)$$

$$R_{transient} = R_s(SOC) e^{-\frac{1}{R_s(SOC)C_s(SOC)}t} + R_m(SOC) e^{-\frac{1}{R_m(SOC)C_m(SOC)}t} + R_h(SOC) e^{-\frac{1}{R_h(SOC)C_h(SOC)}t} \quad (1-10)$$

معادله (1-8) برای SOC را می توان به صورت مدار سمت چپ در شکل ۷-۱ هنگامی که دما و تعداد سیکل ثابت است مدل کرد. اگر معادله (1-8) با ظرفیت باتری نرمالیزه شود، V_{SOC} بین ۰ تا ۱۰۰٪ قرار می گیرد و SOC باتری را نشان می دهد. $I_{battery}$ بار باتری است، در حالی که $R_{self-discharge}$ و $R_{cap-fade}$ به ترتیب دشارژ خودبخود و ضریب تصحیح ظرفیت نرمالیزه شده را نشان می دهند. $V_{terminal}$ به پارامترهای نشان داده شده در مدار سمت راست شکل ۷-۱ مربوط می شود که در آن از رویکرد چند ثابت زمانی (τ_{sec} ، τ_{min} و τ_{hour}) برای مدل کردن رفتار گذرای ولتاژ پایانه استفاده شده است که هر پارامتر نیز تابع SOC می باشد. هر یک از پارامترهای باتری به صورت یک تابع چندجمله ای از SOC حداکثر از مرتبه ۶ بیان می شود:

(1-11)

$$Parameter = a_0 + a_1 SOC + a_2 SOC^2 + \dots$$

برای تعیین ضریب شدت $f_1[i(t)]$ مثلاً در نرخ دشارژ $2C$ (نشان دهنده جریان دشارژی است که ظرفیت نامی باتری را در مدت زمان یک ساعت تخلیه می کند [۱])، ابتدا یک منحنی مرجع برای نرخ دشارژ $2C$ انتخاب می شود که به صورت زیر به دست می آید:

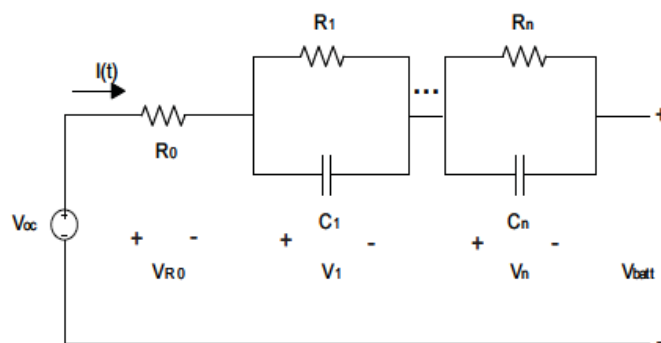
(1-12)

$$2C_{ref} = V_{oc} - R_{int} \times (i_{2C} - i_{C/25})$$

در واقع این منحنی مرجع با حذف افت ولتاژ ناشی از مقاومت داخلی از منحنی دشارژ $C/25$ به دست می آید. اگر متغیرهای a و b به ترتیب معادل با ولتاژ مینیمم در منحنی های دشارژ $2C_{ref}$ و $2C$ باشند، آنگاه نسبت b به a برابر $f_1[i(t)]$ در جریان دشارژ معادل خواهد بود. ضریب دما ($f_2[T(t)]$) را نیز می توان به همان روشی که ضریب شدت $f_1[i(t)]$ به دست آمد پیدا کرد.

مدل مدار معادل رندل^{۲۴}

یکی از مدل‌های الکتریکی که در مقاله‌های زیادی برای تخمین سطح شارژ باتری مورد استفاده قرار گرفته است مدل مدار معادل استاندارد رندل می‌باشد که متشکل از یک منبع ولتاژ ایده‌آل، یک مقاومت R_0 و n شبکه موازی می‌باشد [۲۵]. منبع ولتاژ ایده‌آل معادل ولتاژ مدار باز باتری و R_0 بیان‌گر مقاومت داخلی باتری است. شبکه‌های RC موازی، تأخیر یا هیستریزیس ولتاژ پایانه باتری ناشی از اثر پلاریزاسیون^{۲۵} و دیگر عوامل را شبیه‌سازی می‌کند [۲۶]. از تحلیل ساختار مدل، به این نتیجه می‌رسیم که ساختار با دو شبکه RC در کنار عدم پیچیدگی، ساختاری بهینه هم از لحاظ دقت و هم از لحاظ عملکرد زمان واقعی روی یک میکروپروسسور است [۲۷]. این مدل در شکل ۸-۱ نشان داده شده است.



شکل ۸-۱ مدل مدار معادل رندل [۲۵]

در ادامه معادلات حالت و خروجی حاکم بر این مدار نوشته می‌شود. روابط مربوط به ولتاژها و جریان‌های شبکه‌های RC به صورت زیر می‌باشند:

(۱-۱۳)

$$I_{R_1} + I_{C_1} = \dots = I_{R_n} + I_{C_n} = I$$

(۱-۱۴)

$$\frac{\int I_{C_1} dt}{C_1} = R_1 I_{R_1} = V_1, \dots, \frac{\int I_{C_n} dt}{C_n} = R_n I_{R_n} = V_n$$

که I جریان باتری (مثبت برای دشارژ و منفی برای شارژ) و $V_1, \dots, V_n$ به ترتیب ولتاژ دو سر خازن‌های $C_1, \dots, C_n$ می‌باشند. در حالت کلی R_i, C_i ($i = 1, \dots, n$) و R_0 تابعی از SOC، دما و جریان (مخصوصاً جهت جریان) می‌باشند. با در نظر گرفتن شرایط هم‌دما در دمای اتاق (حذف وابستگی

²⁴ Randle equivalent circuit

²⁵ Polarization

دما)، وابستگی به SOC مینیمم خواهد بود. همچنین وابستگی غیرخطی به اندازه جریان خیلی کوچک است. بنابراین این مقادیر تنها تابعی دودویی^{۲۶} از جهت جریان در نظر گرفته می شوند [۲۵].
از ترکیب روابط (۱-۱۳) و (۱-۱۴) داریم:

(۱-۱۵)

$$I - I_{R_1} = R_1 C_1 \frac{dI_{R_1}}{dt} \quad , \quad \dots \quad , \quad I - I_{R_n} = R_n C_n \frac{dI_{R_n}}{dt}$$

با تبدیل لاپلاس گرفتن از رابطه (۱-۱۵) داریم:

$$G_{I_{R_1}} = \frac{I_{R_1}(s)}{I(s)} = \frac{1}{1 + R_1 C_1 s}$$

⋮

(۱-۱۶)

$$G_{I_{R_n}} = \frac{I_{R_n}(s)}{I(s)} = \frac{1}{1 + R_n C_n s}$$

اگر فرض کنیم که در بازه $[t_k, t_{k+1}]$ ، $dI/dt = 0$ باشد، پاسخ های حالت صفر شبکه های RC به صورت زیر خواهند بود:

$$V_1(s) = R_1 I_{R_1}(s) = \frac{1}{1 + R_1 C_1 s} \cdot \frac{I}{s} \cdot R_1 = \frac{1}{1 + \tau_1 s} \cdot \frac{I}{s} \cdot R_1$$

⋮

(۱-۱۷)

$$V_n(s) = R_n I_{R_n}(s) = \frac{1}{1 + R_n C_n s} \cdot \frac{I}{s} \cdot R_n = \frac{1}{1 + \tau_n s} \cdot \frac{I}{s} \cdot R_n$$

و پاسخ حالت صفر زمان گسسته شبکه های RC به صورت روابط (۱-۱۸) خواهد بود:

$$V_1(k+1) = I_k R_1 \cdot (1 - e^{-\Delta t / \tau_1})$$

⋮

(۱-۱۸)

$$V_n(k+1) = I_k R_n \cdot (1 - e^{-\Delta t / \tau_n})$$

همچنین پاسخ ورودی صفر زمان گسسته شبکه های RC عبارت اند از:

$$V_1(k+1) = V_1(k) \cdot e^{-\Delta t / \tau_1}$$

⋮

(۱-۱۹)

$$V_n(k+1) = V_n(k) \cdot e^{-\Delta t / \tau_n}$$

در اینجا Δt زمان نمونه برداری و $\tau_1, \dots, \tau_n$ به ترتیب ثابت های زمانی برای شبکه های $R_1C_1, \dots, R_nC_n$ هستند، یعنی $R_1C_1 = \tau_1, \dots, R_nC_n = \tau_n$. از جمع روابط (۱۸-۱) و (۱۹-۱)، ولتاژ دو سر خازن های $C_1, \dots, C_n$ در حالت زمان گسسته به دست می آید.

$$\begin{aligned} V_1(k+1) &= V_1(k) \cdot e^{-\Delta t / \tau_1} + I_k R_1 \cdot (1 - e^{-\Delta t / \tau_1}) \\ &\vdots \\ V_n(k+1) &= V_n(k) \cdot e^{-\Delta t / \tau_n} + I_k R_n \cdot (1 - e^{-\Delta t / \tau_n}) \end{aligned} \quad (1-20)$$

عنصر دیگر در این مدل مدار معادل V_{OC} است که تابعی از SOC و دما می باشد (چون شرایط هم دمایی در نظر گرفته می شود وابستگی به دما صرف نظر می شود). وابستگی به دما در دمای ثابت دارای مشخصه های زیر است: (۱) V_{OC} به صورت یکنواخت با دما افزایش می یابد؛ (۲) با نزدیک شدن SOC به $\% 100$ ، V_{OC} با روند نمایی کاهش می یابد؛ (۳) با نزدیک شدن SOC به $\% 100$ ، V_{OC} با روند نمایی افزایش می یابد؛ (۴) در ناحیه بین $\% 20$ و $\% 85$ SOC، رابطه تقریباً خطی می شود. با این مشخصه ها، V_{OC} با یک جفت تابع نمایی به صورت زیر مدل می شود [۲۵]:

$$V_{OC}(SOC) = V_0 + \alpha(1 - e^{-\beta SOC}) + \gamma SOC + \zeta(1 - e^{-\frac{\epsilon}{1-SOC}}) \quad (1-21)$$

که SOC در اینجا به شکل دهدهی^{۲۷} است (یعنی $SOC \in [0,1]$) و $\epsilon, \dots, \beta, \alpha$ پارامترهای ثابت هستند.

چون در اینجا هدف تخمین SOC باتری با استفاده از یک تخمین گر حالت است، SOC باید به عنوان یک حالت در نظر گرفته شود. از روی تعریف SOC در رابطه (۱-۶) خواهیم داشت:

$$\dot{SOC} = -\frac{\eta}{C_n} I \quad (1-22)$$

توجه کنید که این فرمول بندی SOC برای محاسبه SOC واقعی برای داده های آزمایش بعد از پیش پردازش^{۲۸} نیز به کار می رود. از آنجایی که پیش پردازش به طور مؤثری بایاس نویز و کالیبره کردن غلط حسگر را حذف می کند، SOC محاسبه شده از این روش بسیار دقیق است [۲۵].

آخرین جزء مدل یک عنصر هیستریزس است. این عنصر برای به حساب آوردن رفتار باتری در جایی است که ولتاژ حالت استراحت در همان SOC وابسته به تحریک قبلی، متفاوت است [۲۵]. مدل انتخاب شده در اینجا برای توصیف ولتاژ هیستریزس V_h یک معادله دینامیکی مرتبه اول ساده به صورت زیر است:

$$\dot{V}_h = \Gamma |I| (M(SOC, T) - V_h) \quad (1-23)$$

²⁷ Decimal

²⁸ Post-Processing

که Γ ضریب انتقال هیستریزیس (یعنی سرعت رخ دادن هیستریزیس) و M مقدار بیشینه ولتاژ هیستریزیس است که به ازای یک دمای مشخص T رخ می دهد. بر اساس قوانین فیزیکی M باید در حالت شارژ باتری مثبت و در حالت دشارژ منفی باشد. علامت M در وابستگی آن به SOC منعکس می شود که روشی ساده برای بیان جهت جریان است. در نهایت ولتاژ پایانه باتری به صورت زیر بیان می شود:

$$(1-24)$$

$$V_{batt} = V_{OC} - R_0 I - \sum_{i=1}^n V_i - V_h$$

مدل های الکتریکی دیگر

علاوه بر این مدل های مدار الکتریکی که بیان شد، مدل PSPICE و مدل شبکه الکترونیکی²⁹ نیز جزو این مدل ها می باشند [28]. در مدل PSPICE، ولتاژهای سلول، سطح شارژ، وابستگی به نرخ جریان و مشخصه های گرمایی همگی در حلقه های جداگانه مدل می شوند. در این مدل ها از کدهای PSPICE و جداول مراجعه ای³⁰ برای پیکربندی مدل استفاده می شود. مدل شبکه الکترونیکی ترکیبی از مدل الکتروشیمیایی و مدل مدار الکتریکی است. این مدل مدلی ریاضی بر پایه سینماتیک انتقال شارژ و محدودیت های انتقال جرم در واکنش های مختلف می باشد. برای هر فرایند، معادلات ریاضی به دست آمده با عناصر مدار معادل نشان داده می شوند که در یک شبکه الکترونیکی ترکیب می شوند. بخش الکتریکی، بخش شیمیایی و بخش گرمایی در این مدل شبکه الکترونیکی تمییز داده می شوند.

۲-۳-۱- شناسایی پارامترهای باتری

مدل الکتریکی باتری پرکاربردترین مدل باتری در سیستم مدیریت باتری است و بنابراین روال شناسایی پارامترهای این مدل ها (شامل مقاومت، خازن و ...) در این بخش بیان می شود. برای شناسایی این پارامترها، از داده های جمع آوری شده در تست های عملی انجام شده بر روی باتری استفاده می شود. هدف این است که پارامترهای مدل باتری به گونه ای تعیین شوند که اگر به باتری واقعی و مدل باتری یک ورودی (جریان شارژ یا دشارژ) یکسان اعمال شود، خروجی مدل باتری (ولتاژ پایانه باتری) بر خروجی باتری واقعی منطبق باشد. اولین قدم در انجام تخمین پارامترها، انتخاب ساختار مناسب و پس از آن، انتخاب روش تخمین مناسب است که در زیر با جزئیات بیان می گردد.

انتخاب ساختار مدل

مدل های مختلفی برای باتری در بخش قبل معرفی شده است که همان طور که اشاره شد مدل الکتریکی برای استفاده در سیستم مدیریت باتری مناسب است. از این رو، رویکرد کلی این گزارش مبتنی بر شناسایی مدل های الکتریکی است. تنها مساله باقی مانده، انتخاب مرتبه مناسب برای آن است، که مرتبه مدل الکتریکی نیز به تعداد شبکه های RC موجود در آن مربوط می شود. به طور کلی انتخاب مرتبه مدل، یک مساله مصالحه بایاس/واریانس است. برای تعیین تعداد شبکه های RC ساختار انتخابی، دو ویژگی باید مورد توجه قرار گیرد:

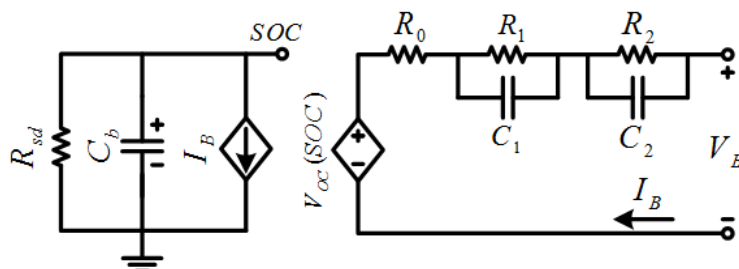
²⁹Electronic-Network

³⁰ Lookup Table

۱- انعطاف پذیری: هر قدر تعداد شبکه های RC برای ساختار بیشتر باشد، سبب افزایش تعداد پارامترهای قابل تعیین و از آنجا باعث کاهش خطا و میزان بایاس موجود بر روی پارامترها خواهد شد.

۲- پیچیدگی: هر قدر تعداد شبکه های RC زیاد شود، پیچیدگی ساختار بیشتر شده و این مساله دو پیامد منفی خواهد داشت. اول آنکه، واریانس پارامترها زیاد خواهد شد و دوم آنکه از مدل های استاندارد قابل پیاده سازی دور خواهد شد.

در [۸۰] دقت مدل های الکتریکی با تغییر مرتبه مدل از ۱ تا n در مقایسه با مدل مرتبه اول تحلیل و بررسی شده است و این نتیجه حاصل شده است که بهبود عملکرد در حالتی که بیش از سه شبکه RC استفاده می شود قابل چشم پوشی است. با توجه به این بررسی ها، برای کاربرد تخمین سطح شارژ در این پروژه، مدل مدار معادل نشان داده شده در شکل ۹-۱ از مرجع [۱] که ترکیبی از مدل رندل مرتبه دوم و مدل زمان-اجرا است مورد استفاده قرار خواهد گرفت. حال معادلات حاکم بر این مدل را می نویسیم.



شکل ۹-۱- مدل انتخابی باتری از مرجع [۱]

در سمت چپ، خازن C_b و منبع جریان کنترل شده با جریان برای مدل سازی ظرفیت و SOC به کار گرفته می شود. سمت راست مدل فوق نیز شامل یک منبع ولتاژ غیرخطی $V_{OC}(SOC)$ به صورت تابعی از SOC و دو شبکه RC برای مدل سازی پاسخ گذرا و یک مقاومت سری برای نشان دادن مقاومت داخلی باتری می باشد. V_B و I_B به ترتیب ولتاژ ترمینال و جریان ترمینال باتری را نشان می دهند. ولتاژ پایانه باتری به صورت زیر بیان می شود:

$$V_B = V_{OC}(SOC) - R_0 I_B - V_1 - V_2 \quad (۱-۲۵)$$

چون در اینجا هدف تخمین SOC باتری با استفاده از یک تخمین گر حالت است، SOC باید به عنوان یک حالت در نظر گرفته شود. SOC به صورت زیر تعریف می شود:

$$SOC(t) = SOC(t_0) - \int_{t_0}^t \frac{I_B}{3600 C_n} d\tau \quad (۱-۲۶)$$

که در آن $SOC(t_0)$ مقدار اولیه SOC، C_n ظرفیت نامی باتری بر حسب آمپر ساعت، I_B جریان لحظه ای (مثبت برای دشارژ و منفی برای شارژ) می باشند. از روی شکل ۹-۱ خواهیم داشت:

$$I_{R_{sd}} + I_{C_b} = -I_B$$

$$\frac{\int I_{C_b} dt}{C_b} = R_{sd} I_{R_{sd}} = SOC \quad (1-27)$$

$$\Rightarrow \dot{SOC} = -\frac{1}{R_{sd}C_b} SOC - \frac{1}{C_b} I_B, C_b = 3600C_n$$

که در آن R_{sd} بیانگر مقاومت دشارژ خودبه خودی است.
روابط مربوط به ولتاژها و جریان های شبکه های RC به صورت زیر می باشند:

$$I_{R_1} + I_{C_1} = I_{R_2} + I_{C_2} = I_B \quad (1-28)$$

$$\frac{\int I_{C_1} dt}{C_1} = R_1 I_{R_1} = V_1, \quad \frac{\int I_{C_2} dt}{C_2} = R_2 I_{R_2} = V_2 \quad (1-29)$$

که I_B جریان باتری (مثبت برای دشارژ و منفی برای شارژ) و V_1, V_2 به ترتیب ولتاژ دو سر خازن های C_1, C_2 می باشند.
از ترکیب روابط (۲-۲۲) و (۲-۲۳)، رابطه ولتاژ دو سر شبکه های RC به صورت زیر به دست می آید:

$$\dot{V}_1 = -\frac{1}{R_1 C_1} V_1 + \frac{1}{C_1} I_B$$

$$\dot{V}_2 = -\frac{1}{R_2 C_2} V_2 + \frac{1}{C_2} I_B \quad (1-30)$$

عنصر دیگر در این مدل مدار معادل، V_{OC} است که تابعی غیرخطی از SOC می باشد. V_{OC} به صورت زیر مدل می شود [۱]:

$$V_{OC}(SOC) = \alpha_0 + \alpha_1 SOC + \alpha_2 SOC^2 + \alpha_3 SOC^3 + \alpha_4 e^{-\beta SOC} \quad (1-31)$$

که در آن $SOC \in [0, 1]$ و $\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \beta$ پارامترهای ثابت هستند.
بنابراین معادلات فضای حالت باتری به صورت زیر خواهد بود:

$$\begin{cases} \dot{SOC} = -\frac{1}{R_{sd}C_b} SOC - \frac{1}{C_b} I_B \\ \dot{V}_1 = -\frac{1}{R_1 C_1} V_1 + \frac{1}{C_1} I_B \\ \dot{V}_2 = -\frac{1}{R_2 C_2} V_2 + \frac{1}{C_2} I_B \end{cases} \quad (1-32)$$

$$V_B = V_{OC}(SOC) - V_1 - V_2 - R_0 I_B$$

انتخاب روش تخمین

یک روش ساده برای استخراج پارامترهای مدل، تست جریان پله ای است [۸]. در ابتدا، منحنی OCV بر حسب SOC اندازه گیری می شود. مقاومت داخلی می تواند با اندازه گیری تغییر لحظه ای ولتاژ در زمان

اعمال جریان پله محاسبه شود و پارامترهای R و C در شبکه‌های RC می‌تواند به طور تجربی با برآزش منحنی پاسخ ولتاژ استخراج شود.

برای شناسایی برخط پارامتر، حداقل مربعات بازگشتی وزن‌دار (RLS) در [۸۲] ارائه شد. معادلات به دست آمده از مدل به یک فرم بازگشتی فرمول‌بندی می‌شود و سپس پارامترها با استفاده از روش RLS شناسایی می‌گردد. این روش فقط به جریان و ولتاژ پایانه باتری به عنوان ورودی نیاز دارد. از جمله روش‌های خارج از خط که برای این منظور به کار گرفته می‌شود می‌توان به روش الگوریتم ژنتیک (GA) [۸۳]، روش بهینه‌سازی ازدحام ذرات (PSO) [۸۵] و روش زیرفضا [۸۶] اشاره کرد.

معیار خطا

تخمین پارامترها، از طریق معرفی یک معیار مناسب، که عموماً به صورت یک تابع اسکالر مثبت معین تعریف می‌شود، و مینیمم سازی آن صورت می‌گیرد. در واقع قرار است یک تابع هزینه بهینه شود، که این تابع هزینه معمولاً تابعی از خطای بین ولتاژ باتری واقعی و ولتاژ مدل باتری است. معیارهای مختلفی برای توصیف خطا وجود دارد که برخی از آن‌ها در روابط (۲-۲۷) تا (۲-۲۹) آمده است. به طوری که $(\hat{y}(t))$ خروجی مدل و $(y(t))$ خروجی واقعی باتری می‌باشد. با آنکه عموماً از معیار (۲-۲۷) استفاده می‌شود، اما همه این معیارها می‌تواند برای تخمین پارامترها استفاده شود.

$$\frac{\sum_{t=1}^N (y(t) - \hat{y}(t | \hat{\theta}_N))^2}{\sum_{t=1}^N y^2(t)} \times 100 \quad (1-33)$$

$$\frac{\sum_{t=1}^N (|y(t)| - |\hat{y}(t | \hat{\theta}_N)|)}{\sum_{t=1}^N |y(t)|} \times 100 \quad (1-34)$$

$$\sum_{t=1}^N \frac{|y(t) - \hat{y}(t | \hat{\theta}_N)|}{|y(t)|} \times 100 \quad (1-35)$$

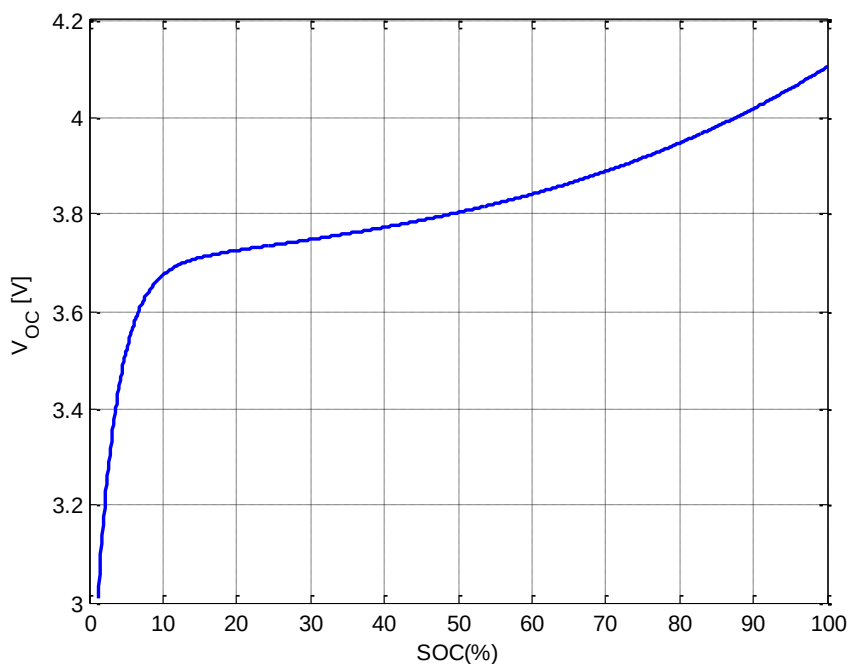
شناسایی پارامترهای مدل انتخابی باتری

خازن ظرفیت کامل C_b موجود در سمت چپ شکل با استفاده از رابطه $C_b = 3600C_n$ به دست می‌آید که C_n ظرفیت باتری بر حسب آمپر ساعت است. مقاومت دشارژ خودبخودی (R_{sd}) یک مقاومت خیلی بزرگ است، اگر اندازه‌گیری جریان دقیق باشد، آنگاه مقاومت R_{sd} طوری به دست می‌آید که SOC به دست آمده از رابطه (۲-۲۱) بر SOC به دست آمده از روش شمارش کلمب (مبتنی بر انتگرالگیری از جریان ترمینال) (رابطه ۲-۲۰) منطبق باشد.

استخراج منحنی Voc بر حسب SOC

Voc می‌تواند به صورت تابعی غیرخطی از SOC بیان شود. طبق تعریف، ولتاژ مدار باز باتری ولتاژ اندازه‌گیری شده در ترمینال باتری است هنگامی که هیچ باری به آن وصل نباشد و تمام فرایندهای شیمیایی داخلی آن به حالت دائمی رسیده باشند. چندین ساعت طول می‌کشد تا باتری به شرایط حالت دائمی برسد. اما نشان داده شده است که برای اهداف عملی، ۱۵-۱۰ دقیقه زمان برای این منظور کافی

است [۷۱]. در یک تست دشارژ یا شارژ باتری با یک جریان پالسی، ولتاژ انتهای زمان استراحت به عنوان ولتاژ مدار باز باتری در نظر گرفته می شود. حال با داشتن مقادیر V_{OC} و نیز SOC به دست آمده از روش شمارش کلمب، منحنی V_{OC} بر حسب SOC استخراج می شود. در نهایت با برازش کردن منحنی رابطه (۱-۳۱) بر منحنی V_{OC} بر حسب SOC، پارامترهای این منحنی به دست می آیند. منحنی نوعی V_{OC} SOC در یک باتری لیتیم-یون به صورت شکل زیر است.



شکل ۱۰-۱- منحنی V_{OC} بر حسب SOC

تخمین مقاومت داخلی R_0

برای محاسبه R_0 از نسبت تغییرات ولتاژ ناشی از پرش جریان استفاده می کنیم [۷۹]. مقدار R_0 در i امین نقطه پرش جریان می تواند به صورت زیر بیان شود:

$$R_{0,i} = \frac{V_{\max,i} - V_{\min,i}}{i_{\max,i} - i_{\min,i}} \quad (1-36)$$

که $V_{\max,i}$, $V_{\min,i}$, $i_{\max,i}$, $i_{\min,i}$ مقادیر ولتاژ و جریان قبل و بعد از پرش ناگهانی است. مقدار میانگین $R_{0,i}$ در چند نقطه پرش جریان، R_0 را می دهد:

$$R_0 = \sum_{i=1}^n R_{0,i} / n \quad (1-37)$$

تخمین پارامترهای شبکه های RC

در [۸۴]، از مدل دینامیکی ARMAX برای تخمین پارامترهای شبکه های RC استفاده می شود. ساختار کلی این مدل به صورت رابطه زیر است، که در آن u_t ورودی سیستم و ℓ_t نویز سفید است.

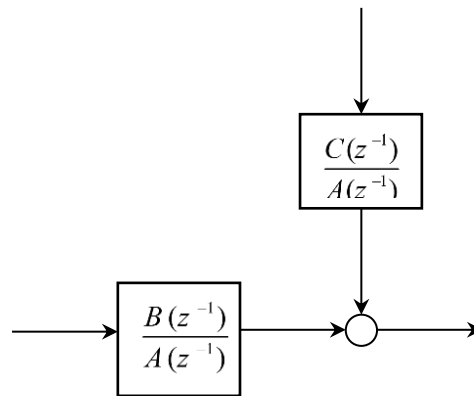
$$y_t = \frac{B(z^{-1})}{A(z^{-1})} u_t + \frac{C(z^{-1})}{A(z^{-1})} e_t \quad (1-38)$$

به طوری که:

$$A(z^{-1}) y_t = 1 + a_1 z^{-1} + \dots + a_{na} z^{-na} \quad (1-39)$$

$$B(z^{-1}) y_t = b_1 z^{-1} + \dots + b_{nb} z^{-nb} \quad (1-40)$$

$$C(z^{-1}) = 1 + c_1 z^{-1} + \dots + c_{nc} z^{-nc} \quad (1-41)$$



شکل ۱-۱۱- ساختار مدل ARMAX

برای استخراج مدل ARMAX از باتری، فرم لاپلاس ولتاژ پایانه باتری به صورت زیر به دست می آید:

$$\begin{aligned} V_B &= V_{OC}(SOC) - \left(R_0 + \frac{R_1}{1 + R_1 C_1 s} + \frac{R_2}{1 + R_2 C_2 s} \right) I_B \\ &= V_{OC}(SOC) - R_0 I_B - G(s) I_B \end{aligned} \quad (1-42)$$

رابطه (۱-۴۲) را به فرم زیر می نویسیم:

$$V_{RC}(t) = -V_B(t) + V_{OC}(SOC) - R_0 I_B(t) = G(s) I_B(t) \quad (1-43)$$

با فرض $di_B/dt \approx 0$ در یک زمان نمونه برداری، رابطه فوق به صورت زیر گسسته می شود:

$$(1-44)$$

$$V_{RC}[k] = \left(\frac{R_1(1 - e^{-T_s/T_{p1}})z^{-1}}{1 - e^{-T_s/T_{p1}}z^{-1}} + \frac{R_2(1 - e^{-T_s/T_{p2}})z^{-1}}{1 - e^{-T_s/T_{p2}}z^{-1}} \right) I_B[k] = \left(\frac{z^{-1}(b_1 + b_2 z^{-1})}{1 + a_1 z^{-1} + a_2 z^{-2}} \right) I_B[k]$$

که در آن T_s زمان نمونه برداری است. حال پارامترهای a_2, a_1, b_2, b_1 تخمین زده می شوند و از روی آن ها پارامترهای شبکه های RC به دست می آیند.

۳-۱- روش های موجود برای تخمین SOC

سیستم های مدیریت باتری (BMS) جزء کلیدی وسایل نقلیه الکتریکی ترکیبی (HEV) می باشند و از طرفی دانستن حالت باتری ها مانند SOC که وظیفه اصلی BMS می باشد ضروری است. لذا BMS باید قادر به تخمین برخط SOC موجود باتری به صورت دقیق باشد. همان طور که بیان شد، SOC به صورت نسبت آمپر ساعت موجود باتری به آمپر ساعت کل که در زمان شارژ کامل قابل دسترسی است تعریف می شود. تاکنون مجموعه ای از روش ها برای اندازه گیری یا تخمین SOC ارائه شده است که در [۳۳] به طور خلاصه آمده است و هر کدام دارای مزیت و معایبی است. در میان این روش ها، تعدادی مانند روش های مبتنی بر فیلتر کالمن، در BMS با موفقیت مورد استفاده قرار گرفته اند.

در حالت کلی، دو دسته روش برای تخمین SOC وجود دارند: روش های مستقیم و غیرمستقیم. در روش های غیرمستقیم، SOC از روی برخی خواص فیزیکی باتری مانند چگالی اسید یا پالس های گالوانواستاتیک کاتدی تخمین زده می شود. تخمین این کمیت ها نیاز به اندازه گیری های دقیق دارد و در عمل دارای محدودیت هایی است [۳۴-۳۵]. یکی از روش های غیرمستقیم، اندازه گیری ولتاژ مدار باز باتری می باشد. ثابت شده است که ولتاژ مدار باز تثبیت شده با وابستگی ناچیز به دما یا وضعیت گذشته باتری، SOC باتری را به طور دقیقی نشان می دهد. در این روش باتری باید برای مدتی در استراحت باشد تا اجازه دهد ولتاژ مدار باز به حالت دائمی برسد. در کاربردهایی که باتری برای چندین ساعت به طور پیوسته کار می کند و باتری نمی تواند از مدار باز شود، این روش عملی نیست [۳۵-۳۷]. در روش های دیگر، SOC با استفاده از ولتاژ دشارژ باتری تخمین زده می شود [۳۸-۳۹]. اساس این روش، رابطه بین ولتاژ دشارژ و SOC می باشد. این رابطه نباید از شرایط دشارژ مانند نرخ دشارژ، دمای محیط و SOC اولیه تأثیر پذیرد. در حوزه زمان ولتاژ دشارژ باتری تابعی غیرخطی از شرایط دشارژ است [۳۸]. توابع نمایی، سهموی و هذلولی برای بیان این رابطه غیرخطی استفاده شده اند. اما این رابطه مقاوم نیست، چون واگرایی قابل ملاحظه ای هنگام تغییر شرایط دشارژ مشاهده می شود. برای مقاوم تر کردن نمایش دشارژ، ولتاژ دشارژ در حوزه SOC بیان می شود. این رابطه برای محدوده وسیعی از نرخ های دشارژ و دماهای محیط معتبر است. به دلیل طبیعت مقاوم این رابطه، تنها یک مشخصه دشارژ تحت شرایط کاری نامی برای مدل کافی است [۳۸]. در [۳۹] فقط زمان باقی مانده برای رسیدن به ولتاژ انتهای دشارژ به کمک شبکه های عصبی مصنوعی (ANN)^{۳۱} تخمین زده می شود. طیف نمایی امپدانس یک روش غیرمستقیم مرسوم برای فرایندهای الکتروشیمیایی مانند باتری است. این روش نه تنها برای تخمین SOC، بلکه برای تخمین SOH نیز به کار می رود [۴۰-۴۱]. اما این رویکرد نیاز به اندازه گیری های اضافی دارد که این روش را برای تست های آزمایشگاهی مناسب می سازد نه در کاربردهای عملی [۳۵-۳۶]. در [۴۲]، امپدانس باتری به طور مستقیم از طریق تغییرات فرکانس برای بهبود فرایند شارژ اندازه گیری می شود. در [۴۳]، SOC از روی تخمین ولتاژ نیروی الکتروحرکتی (EMF)^{۳۲} تعیین می شود که ولتاژ EMF بیانگر جمع پتانسیل تعادل دو الکتروود باتری می باشد که همان ولتاژ مدار باز باتری است. رابطه بین ولتاژ EMF و SOC در زیر آمده است.

³¹ Artificial Neural Network

³² Electromotive Force

$$V_{EMF} = \theta * A_{\theta} + EMF_{\min} \quad (1-45)$$

که θ همان SOC است، A_{θ} شیب تغییر ولتاژ EMF با SOC و $EMF_{\min}$ مقدار کمینه EMF در SOC صفر است. این رابطه می تواند بر حسب SOC بازنویسی شود.

$$\theta = \frac{V_{EMF} - EMF_{\min}}{A_{\theta}} \quad (1-46)$$

$EMF_{\min}$ و A_{θ} در کاتالوگ های^{۳۳} کارخانه سازنده موجود است.

برخی محققان از منطق فازی برای مدل کردن رابطه بین SOC باتری و پارامترهای آن بر مبنای اندازه گیری طیف امپدانس استفاده کرده اند [۴۴-۴۵]. با اندازه گیری امپدانس در سه فرکانس، مدل منطق فازی قادر به تخمین هر دو کمیت ظرفیت باقیمانده و SOC خواهد بود. این مدل، SOC را با خطای کمتر از ۵٪ تخمین می زند.

روش شمارش کلمب (شمارش آمپرساعت)^{۳۴} یا انتگرال گیری جریان، مرسوم ترین روش تخمین SOC می باشد که نیاز به اندازه گیری جریان باتری دارد.

$$SOC(t) = SOC(t_0) - \int_{t_0}^t \frac{\eta I}{C_n} d\tau \quad (1-47)$$

که در آن $SOC(t_0)$ مقدار اولیه SOC، C_n ظرفیت نامی باتری بر حسب آمپرساعت، I جریان لحظه ای (مثبت برای دشارژ و منفی برای شارژ) و η ضریب کلمب ($\eta = 1$ برای دشارژ و $\eta \leq 1$ برای شارژ) می باشند.

به دلیل مسائلی مانند مسأله مقدار اولیه، خطای انباشته شده ناشی از اندازه گیری های نادقیق جریان و همچنین در نظر نگرفتن تغییرات ظرفیت باتری با نرخ دشارژ، دما و اتلاف ضریب کلمب، این روش در زمان واقعی قابل اعتماد نیست. برای بهبود روش شمارش کلمب، روش ترکیبی شمارش کلمب و ولتاژ مدار باز^{۳۵} (V_{OC}) به کار گرفته می شود. در این روش، V_{OC} اندازه گیری شده برای تصحیح پی در پی SOC به دست آمده از روش شمارش کلمب استفاده می شود. اما همانطور که بیان شد، V_{OC} فقط بعد از اینکه باتری برای یک مدت زمان نسبتاً طولانی (اغلب چندین ساعت) در استراحت به سر برد، می تواند اندازه گیری شود و لذا این روش عملی نیست [۲۷-۲۵].

شبکه های عصبی مصنوعی توسط برخی محققان برای تخمین مستقیم SOC به کار گرفته شده اند [۵۵-۶۲]. روش های مبتنی بر شبکه عصبی مصنوعی (ANN)^{۳۶} [۸۷] از نگاشت غیرخطی برای تولید مدل باتری استفاده می کنند و تخمین SOC مستقیماً از روی مشخصه نگاشت غیرخطی مشابه روش تخمین مستقیم SOC در روش مبتنی بر مدل انجام می شود. در این روش ها نیاز به برخی داده های ورودی-خروجی به عنوان داده های آموزش می باشد که باید با روش های تخمین دیگر به دست آید. سپس شبکه آموزش داده شده می تواند برای تخمین SOC استفاده شود. در [۵۴] یک مدل فضای حالت SOC به

³³datasheet

³⁴ Coulomb counting (Ampere-hour counting)

³⁵ Open Circuit Voltage

³⁶ Artificial Neural Network

کمک شبکه عصبی^{۳۷} (NN) تقریب زده می شود که متغیرهای حالت این مدل، ولتاژ پایانه باتری در لحظه قبلی و SOC در این لحظه می باشند. سپس با استفاده از این مدل و EKF، SOC باتری تخمین زده می شود. آموزش شبکه عصبی به صورت خارج از خط^{۳۸} و با استفاده از داده هایی که از فرایند شارژ باتری جمع آوری شده است صورت می گیرد. چنگ-هوی کای، یک سیستم استنتاج نروفازی تطبیقی (ANFIS) جهت تخمین و پیش بینی وضعیت شارژ باتری ارائه کرده است [۶۳]. روش ANFIS از مزیت هر دو روش شبکه عصبی و روش فازی استفاده می کند تا دقت تخمین SOC را بهبود بخشد. ورودی های شبکه، ولتاژ، جریان و دمای سطح سلول بوده و SOC تنها خروجی آن است.

برای بهبود دقت تخمین SOC مبتنی بر مدل، رؤیتگرهای تطبیقی از جمله رؤیتگر مد لغزشی (SMO)^{۳۹} و فیلتر کالمن توسعه یافته (EKF) پیشنهاد شده اند. با استفاده از این روش ها، تخمین حلقه بسته SOC پیاده سازی می شود، که در آن اختلاف بین ولتاژ اندازه گیری باتری و ولتاژ مدل برای تصحیح تخمین استفاده می شود.

۴-۳-۱- تخمین سطح شارژ با استفاده از فیلتر کالمن

فیلتر کالمن (KF) معروف که در طی سال های ۱۹۶۰ توسعه داده شد، الگوریتمی بازگشتی برای تخمین حالت های داخلی سیستم های دینامیکی ارائه می دهد. خاصیت یکتای KF این است که حالت های متأثر از نویز باند پهن که در پهنای باند سیستم قرار دارد را تخمین می زند و یک مصالحه تجربی بین خطای مدل سازی و اثر نویز برقرار می کند. تعدادی محقق از این فیلتر برای تخمین ولتاژ مدار باز یا پارامترهای دیگر باتری که رابطه مستقیمی با SOC دارند استفاده کرده اند [۴۶-۴۸]. در [۴۹]، KF برای تخمین برخی کمیت های فیزیکی که تأثیر مستقیمی روی SOC دارند به کار گرفته شده است. در برخی مقاله ها، SOC به عنوان حالت مدل تعریف شده و با استفاده از KF تخمین زده می شود [۵۰-۵۳]. در الگوریتم فیلتر کالمن، تخمین بر اساس یک مدل از دینامیک سیستم انجام می شود و لذا KF نیاز به مدل مناسبی از باتری دارد. مدل به کار رفته برای باتری در اغلب روش های مبتنی بر فیلتر کالمن، مدل مدار معادل رندل مرتبه دوم می باشد که معادلات حاکم بر آن نیز آورده شده است. همچنین به دلیل استفاده از فیدبک در این فیلتر، نیاز به مقداردهی اولیه مناسب حالت ها می باشد؛ در غیر این صورت حالت ها ممکن است همگرا نشوند [۵۴].

الگوریتم فیلتر کالمن توسعه یافته

اگر سیستم غیرخطی باشد، باید با استفاده از یک فرایند خطی سازی در هر گام نمونه برداری، سیستم غیرخطی با یک سیستم خطی متغیر با زمان تقریب زده شود. فیلتر کالمنی (KF) که حالت های این مدل خطی متغیر با زمان را تخمین می زند، فیلتر کالمن توسعه یافته (EKF)^{۴۰} نامیده می شود [۲۶ و ۵۴]. اگرچه عملکرد EKF بهینه نیست، برای اغلب کاربردها خوب کار می کند [۲۶]. فرض می کنیم که سیستم غیرخطی زیر را داریم:

³⁷Neural Network

³⁸ Offline

³⁹ Sliding Mode Observer

⁴⁰ Extended Kalman Filter

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k \\ \mathbf{y}_k &= \mathbf{g}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{v}_k \end{aligned} \quad (1-48)$$

که در آن $\mathbf{f}(\cdot, \cdot)$ تابع انتقال حالت، $\mathbf{g}(\cdot, \cdot)$ تابع اندازه گیری، $\mathbf{w}_k$ نویز فرایند و $\mathbf{v}_k$ نویز اندازه گیری هستند. این نویزها، نویزهای گوسی سفید مستقل با میانگین صفر و با ماتریس های کواریانس زیر می باشند:

$$\mathbf{Q}_k = E[\mathbf{w}_k \mathbf{w}_k^T], \quad \mathbf{R}_k = E[\mathbf{v}_k \mathbf{v}_k^T] \quad (1-49)$$

یک موضوع مهم در طراحی KF، انتخاب مناسب ماتریس کواریانس برای نویز اندازه گیری و فرایند می باشد [54]. ماتریس کواریانس نویز اندازه گیری ($\mathbf{R}_k$) می تواند به طور مستقیم از داده های باتری تعیین شود. واریانس ها را می توان از مربع ریشه میانگین مربعات (rms) ⁴¹ اندازه گیری های نویزی ولتاژ ترمینال باتری به دست آورد. فرض می شود که واریانس ها مستقل هستند و توزیع گوسی دارند. برای اعمال EKF به مدل غیرخطی ارائه شده در رابطه (1-48)، مدل باتری باید در هر زمان نمونه برداری با استفاده از بسط سری تیلور حول نقطه کار خطی شود. مشتق گیری از توابع $\mathbf{f}(\cdot, \cdot)$ و $\mathbf{g}(\cdot, \cdot)$ نسبت به $\mathbf{x}_k$ نتیجه می دهد:

$$\mathbf{A}_k = \left. \frac{\delta \mathbf{f}}{\delta \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_k} = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & \alpha_1 & \dots & 0 & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & \alpha_n & 0 \\ 0 & 0 & 0 & 0 & \gamma(I_k) \end{bmatrix} \quad (1-50)$$

$$\mathbf{C}_k = \left. \frac{\delta \mathbf{g}}{\delta \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_k^-} = \left[\left. \frac{dV_{OC}}{dSOC} \right|_{SOC=\hat{SOC}_k^-}, -1, \dots, -1, -1 \right] \quad (1-51)$$

در روابط فوق $\hat{\mathbf{x}}_k$ به عنوان پس تخمین حالت $\mathbf{x}_k$ بعد از زمان k تعریف شده است که بادر دست داشتن اندازه $\mathbf{y}_k$ قابل محاسبه است. همچنین $\hat{\mathbf{x}}_k^-$ به عنوان پیش تخمین حالت در زمان k تعریف شده است که با در دست داشتن اطلاعات از فرایند قبل از زمان k قابل محاسبه است. حال با در نظر گرفتن مقادیر اولیه زیر برای متغیرهای حالت و کواریانس خطا، الگوریتم EKF به صورت جدول ۱-۱ خلاصه می شود.

$$\begin{aligned} \hat{\mathbf{x}}_0 &= E[\mathbf{x}_0] \\ \mathbf{P}_0 &= E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T] \end{aligned} \quad (1-52)$$

جدول ۱-۱- خلاصه الگوریتم EKF [26 و 54]

⁴¹Root Mean Square

$k = 1, 2, \dots$	محاسبه به ازای
$\hat{\mathbf{x}}_k^- = \mathbf{f}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1})$	به روزرسانی زمان
$\mathbf{P}_k^- = \mathbf{A}_{k-1} \mathbf{P}_{k-1} \mathbf{A}_{k-1}^T + \mathbf{Q}_{k-1}$	انتشار تخمین حالت
$\mathbf{L}_k = \mathbf{P}_k^- \mathbf{C}_k^T (\mathbf{C}_k \mathbf{P}_k^- \mathbf{C}_k^T + \mathbf{R}_k)^{-1}$	انتشار کواریانس خطا
$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{L}_k (\mathbf{y}_k - \mathbf{g}(\hat{\mathbf{x}}_k^-, \mathbf{u}_k))$	به روزرسانی اندازه گیری
$\mathbf{P}_k = (\mathbf{I} - \mathbf{L}_k \mathbf{C}_k) \mathbf{P}_k^-$	ماتریس بهره کالمن
	به روزرسانی تخمین حالت
	به روزرسانی کواریانس خطا

الگوریتم EKF در مقاله های بسیاری برای تخمین SOC به کار رفته است. روش های مبتنی بر فیلتر کالمن (KF) [۸۸] و فیلتر کالمن توسعه یافته (EKF) [۸۹]، جزو پرکاربردترین روش ها در تخمین SOC می باشد. در [۲۶] مدل مدار معادل رندل با دو شبکه RC به کار رفته است. در [۲۷] با اضافه کردن مقاومت داخلی باتری به عنوان یک متغیر حالت جدید، اصلاح زمان واقعی واریانس خطای مشاهده R_k و اضافه کردن ضریب بهره برای کنترل سرعت همگرایی، روش EKF ارائه شده در [۲۶] بهبود داده شده است. در [۲۵] یک تخمین گر حالت مبتنی بر مدل با استفاده از تکنیک سیستم خطی متغیر با پارامتر (LPV)^{۴۲} طراحی شده است که رویکرد محاسباتی کم برای EKF فراهم می کند. در [۹۰] از یک KF تطبیقی (AKF) برای تخمین SOC از روی یک مدل RC مرتبه دو باتری استفاده شده است. برخلاف KF معمولی که در آن میانگین و کوواریانس نویز مدل و نویز اندازه گیری ثابت فرض می شود، در این مقاله، از یک الگوریتم تطبیق مبتنی بر یک ضریب فراموشی برای استخراج برخط این ماتریس ها استفاده می شود. این امر باعث افزایش نرخ ردیابی مقدار واقعی SOC می شود. در [۹۱] از یک EKF تطبیقی (AEKF) برای تخمین برخط حالت ها و پارامترهای مدل الکتریکی باتری استفاده می شود. تمام پارامترهای مدل باتری به عنوان متغیر حالت در معادلات حالت سیستم در نظر گرفته می شوند. در این الگوریتم، ماتریس کوواریانس جدیدی از روی بردار خطای تخمین خروجی در یک پنجره تخمین متحرک ساخته می شود و ماتریس های کوواریانس نویز و فرایند از روی آن تخمین زده شده و بروز می شوند. در [۹۲] یک روش مبتنی بر EKF مدل تطبیقی (MAEKF)^{۴۳} برای تخمین SOC بر مبنای یک مدل الکتریکی ارائه می شود. تحلیل حساسیت مدل الکتریکی نشان می دهد که دقت SOC تخمین زده شده با EKF به مقاومت های مدل حساس است. برای بهتر کردن تخمین، مقادیر مقاومت ها در یک فرایند

⁴²Linear Parameter Varying⁴³ Model Adaptive EKF

بهینه سازی به دست می آیند. در [۹۳] یک EKF چندمقیاسه (MEKF)^{۴۴} برای تخمین SOC و پارامتر باتری به کار گرفته شده است. با توجه به مشخصه تغییرات کند برای پارامتر و مشخصه تغییرات تند برای SOC، پارامتر با مقیاس ماکرو و SOC با مقیاس میکرو تخمین زده می شود.

۵-۳-۱- تخمین سطح شارژ با استفاده از رؤیتگر مد لغزشی

رؤیتگر مد لغزشی، توسعه یافته ی کنترل مد لغزشی در تخمین حالت با رؤیتگر می باشد. در اینجا نیز مسیر SMO باید در مدت زمان محدود بر روی یک سطح لغزشی مناسب قرار گیرد [۷۲-۷۳]. برای تخمین حالت با رؤیتگر مد لغزشی، معادلات حالت سیستم به صورت زیر نوشته می شود:

$$\begin{cases} \dot{x} = Ax + Bu + Tf(x, u) \\ y = Cx \end{cases} \quad (۱-۵۳)$$

که در آن $A \in \mathbb{R}^{n \times n}$ ، $B \in \mathbb{R}^{n \times m}$ ، $C \in \mathbb{R}^{p \times n}$ و $T \in \mathbb{R}^{n \times q}$ که $p \geq q$ و ماتریس های T ، C دارای رتبه کامل فرض می شوند [۷۴]. ترم اغتشاشی $Tf(x, u)$ برای نشان دادن خطاهای مدل سازی و خطی سازی به کار می رود. پارامتر T ماتریس ورودی اغتشاشی و تابع $f \in \mathbb{R}^q$ محدود فرض می شود، $\|f\| \leq h_0$ ، که h_0 ثابت است.

ایده اصلی SMO این است که رؤیتگری طراحی شود که بتواند تخمین حالت $\hat{x}$ و تخمین خروجی $\hat{y} = C\hat{x}$ را طوری به دست آورد که خطای تخمین حالت $e = x - \hat{x}$ وقتی $t \rightarrow \infty$ ، به صفر میل کند. این رؤیتگر به صورت زیر داده می شود:

$$\begin{cases} \dot{\hat{x}} = A\hat{x} + Bu + H(y - \hat{y}) + hTv(e_y) \\ \hat{y} = C\hat{x} \end{cases} \quad (۱-۵۴)$$

که در آن ماتریس بهره رؤیتگر H و بهره سوئیچینگ h طوری انتخاب می شوند که پایداری SMO تضمین شود. ترم $hTv(e_y)$ قاعده سوئیچینگ ناپیوسته را نشان می دهد که در آن $v(e_y)$ سیگنال تزریق خطای خروجی $e_y = y - \hat{y}$ می باشد و به صورت زیر تعریف می شود:

$$v(e_y) = \text{sgn}(Fe_y) = \begin{cases} +1 & e_y > 0 \\ -1 & e_y < 0 \end{cases} \quad (۱-۵۵)$$

که در آن ماتریس F یک ماتریس مثبت معین می باشد که در ادامه معرفی می شود. ماتریس بهره رؤیتگر H با استفاده از روش جایابی قطب تعیین می شود. شرط لازم و کافی وجود بهره رؤیتگر H برای جایابی قطب های رؤیتگر، رؤیت پذیری سیستم است.

به ازای ماتریس مثبت معین Q ، یک جواب منحصر به فرد مثبت معین P برای معادله لیاپانوف زیر وجود دارد:

$$P(A - HC) + (A - HC)^T P = -Q \quad (۱-۵۶)$$

حال فرض می کنیم که به ازای ماتریس مثبت معین P ، شرط زیر به ازای یک ماتریس مثبت معین F برآورده می شود:

⁴⁴ Multi-scale EKF

$$FC = T^T P \quad (1-57)$$

این ماتریس F ، همان ماتریس معرفی شده در رابطه (1-55) می باشد. در [76] نشان داده شده است که شروط لازم و کافی برای وجود ماتریس های P و F که در روابط (1-56) و (1-57) صدق کنند و در نتیجه یک حرکت لغزشی در زمان محدود بر روی سطح $S = \{e \in R^n : e_y = C e = 0\}$ صورت گیرد، عبارتند از:

$$rank(CT) = q \quad 1.$$

$$صفرهای نامتغیر (A, T, C) \text{ در نیمه چپ صفحه موهومی قرار گیرد.} \quad 2.$$

رؤیتگر مد لغزشی (SMO)⁴⁵، یکی دیگر از رؤیتگرهای مبتنی بر مدل است که نسبتاً دقیق بوده و نسبت به عدم قطعیت در پارامترهای مدل و اغتشاش مقاوم است. قلی زاده و سلماسی در [94] یک رؤیتگر مد لغزشی برای یک مدل RC مرتبه دو باتری لیتیم-یون با فرض معلوم بودن حد بالای عدم قطعیت مدل سازی ارائه کرده اند و سپس الگوریتمی برای تخمین این عدم قطعیت به کار گرفته اند. در مقایسه با روش های قبلی SMO، مدل به کار رفته برای باتری در این مقاله رؤیت پذیر نیست و این شرط به شرط ضعیف تر آشکارپذیری کاهش یافته است. با استفاده از یک روش جداسازی، حالت های رؤیت ناپذیر و حالت های رؤیت پذیر جدا شده و سپس روش SMO برای تخمین حالت های رؤیت پذیر به کار گرفته می شود. با استفاده از روش لیاپانوف اثبات شده است که خطای تخمین این رؤیتگر به طور مجانبی به صفر میل خواهد کرد. در [95] یک SMO با بهره سوئیچینگ تطبیقی پیشنهاد شده است. در مقایسه با SMO معمولی، این رؤیتگر می تواند سطح لغزش⁴⁶ در تخمین SOC را با استفاده از یک بهره سوئیچینگ خودتطبیق کاهش دهد، در حالی که مشخصه قابلیت جبران خطاهای مدل سازی ناشی از تغییرات پارامترهای مدل را ثابت نگه می دارد. در [96] یک رؤیتگر مد لغزشی تطبیقی بر مبنای یک مدل مدار معادل الکتریکی باتری طراحی گردیده است. پارامتر C_b در مدل باتری بیانگر خازن بدنه⁴⁷ است که توانایی باتری برای ذخیره شارژ را مشخص می کند و SOC باتری از روی آن تعیین می شود. بر مبنای تئوری لیاپانوف یک قاعده تطبیق برای این پارامتر به منظور در نظر گرفتن کاهش ظرفیت باتری در طول کارکرد آن استخراج شده است. با وجود مقاوم بودن روش SMO، پیاده سازی آن آسان نبوده و همچنین پارامترهای آن برای حالت فعلی باتری بروز نمی شوند.

جدول ۱-۲- روش های موجود برای تخمین SOC [33 و 64]

تکنیک	حوزه کاربرد	مزایا	معایب
تست دشارژ	کاربرد برای تعیین ظرفیت در شروع عمر	ساده و دقیق؛ مستقل از SOH	خارج از خط، زمان بر، حالت باتری را تغییر می دهد، اتلاف انرژی
شمارش کلمب	تمام سیستم های باتری، اغلب کاربردها	دقیق اگر نقاط کافی برای کالیبراسیون مجدد موجود باشد و	حساس به عکس العمل نویز اندازه گیری؛ نیاز به نقاط

⁴⁵ Sliding Mode Observer

⁴⁶ Chattering

⁴⁷ Bulk Capacitance

ولتاژ مدار باز	سرب، لیتیم، Zn/Br	ارزان، تخمین Voc	نیاز به زمان استراحت طولانی (جریان=0A)
طیف‌نمایی امپدانس	تمام سیستم‌های باتری	اطلاعاتی در مورد SOH می‌دهد	حساس به دما، هزینه‌بر
مقاومت داخلی DC	سرب، نیکل-کادمیم	اطلاعات در مورد SOH می‌دهد	دقت خوب فقط برای یک بازه زمانی کوتاه
شبکه‌های عصبی	تمام سیستم‌های باتری	برخط	نیاز به داده‌های آموزش از یک باتری مشابه، گران برای پیاده‌سازی
منطق فازی	تمام سیستم‌های باتری	برخط، مقاوم	نیاز به مقدار زیاد حافظه برای کاربردهای زمان واقعی
فیلترهای کالمن	تمام سیستم‌های باتری، حتی کاربردهای	دقیق، انعطاف‌پذیر، برخط	نیاز به به اطلاعات دقیق بر روی نویز فرایند و اندازه‌گیری، نیاز به محاسبات
رُیتگر مد لغزشی	تمام سیستم‌های باتری	دقیق، مقاوم به نویز و خطاهای مدل و مقدار اولیه، دینامیکی	عدم پیاده‌سازی آسان، خطای زیاد در دماهای پایین

در جدول ۲-۱ مروری بر روش‌های ارائه شده در مقالات برای تخمین SOC آورده شده است [۳۳] و [۶۴]. تعدادی از این روش‌ها در کاربرد HEV امکان‌پذیر نیستند، زیرا نیاز به جدا کردن باتری از مدار دارند. تعدادی دیگر که مبتنی بر اندازه‌گیری ولتاژ هستند، برای کاربردهای الکترونیکی توان کم مناسب‌تر هستند که توان مورد نیاز معمولاً ثابت و کوچک می‌باشد. در چنین مواردی، یک ترسیم دشارژ ولتاژ می‌تواند به آسانی برای تخمین SOC به کار رود. معایب این روش این است که نیاز به یک بازه طولانی جریان ثابت دارد که در کاربرد HEV مرسوم نیست. به کار بردن روش‌های هوشمند مانند شبکه‌های عصبی برای داشتن یک تخمین خوب از SOC معمولاً امکان‌پذیر است. اما این روش‌ها نیاز توان محاسباتی بالا دارد که معمولاً در سیستم‌های یکپارچه موجود نیستند. از طرفی، به حجم وسیعی داده برای آموزش شبکه نیاز است و خطای تخمین به طور قابل ملاحظه‌ای تحت تأثیر داده و روش آموزش است. همچنین شبکه آموزش داده شده برای یک باتری نو نمی‌تواند برای یک باتری کهنه دقیق باشد. روش شمارش کلمب هنوز پرکاربردترین روش است، زیرا یک روش ساده برای تخمین SOC فراهم می‌کند. عیب این روش این است که نمی‌توان یک تخمین از SOC اولیه داشت و هر خطایی روی اندازه‌گیری جریان می‌تواند روی تخمین اثر بگذارد. در روش‌های تطبیقی مانند فیلتر کالمن و رُیتگر مد لغزشی، تخمین حلقه بسته SOC پیاده‌سازی می‌شود، که در آن اختلاف بین ولتاژ اندازه‌گیری باتری و ولتاژ مدل برای تصحیح تخمین استفاده می‌شود. خاصیت حلقه بسته در این روش‌ها باعث بهبود دقت تخمین SOC خواهد شد. با وجود حجم محاسبات بالای این روش‌های تطبیقی، دقت، مقاوم بودن، انعطاف‌پذیری و قابلیت تصحیح بر خطا به دلیل خاصیت حلقه بسته این روش‌ها باعث شده است که جزو پرکاربردترین روش‌ها در تخمین SOC باشند. بنابراین در این پروژه نیز از این دو روش برای تخمین SOC باتری استفاده خواهد شد.

۴-۱- روش‌های یکسان‌سازی شارژ باتری

استک‌های باتری در خودروهای الکتریکی و همچنین الکتریکی ترکیبی، معمولاً از تعداد زیادی سلول که به طور سری به هم متصل شده‌اند برای ایجاد منبعی با ولتاژ ۴۰۰-۳۰۰ ولت تشکیل می‌شوند [۹۷] و

[۹۸]. محدودیت های سطح ولتاژ در سلول های لیتیوم-یون وجود یک سیستم مدیریت شارژ و دشارژ در این باتری ها را اجتناب ناپذیر می کند. باتری های لیتیوم-یون نمی توانند ولتاژی بالاتر از حدود $4/3$ ولت را تحمل کنند. در صورتی که ولتاژ این سلول ها از این مقدار بالاتر رود، دمای باتری و به دنبال آن فشار داخل سلول افزایش یافته و می تواند در مواردی منجر به آتش سوزی در استک باتری شود [۱۰۱]. از سوی دیگر، این باتری ها برای اینکه همواره بتوانند ظرفیت خود را حفظ کنند، نباید تا ولتاژی کمتر از $2/4$ ولت تخلیه شوند [۱۰۲]. در این صورت تغییراتی بازگشتناپذیر در ساختار شیمیایی سلول به وجود آمده که منجر به کاهش ظرفیت همیشگی سلول خواهد شد. از طرفی وجود تفاوت هایی در خصوصیات شیمیایی سلول ها و تفاوت های معمول در سیکل های مکرر شارژ-دشارژ، منجر به ایجاد ناهمبستگی در سطح شارژ سلول ها و بنابراین ولتاژهای ترمینال مختلف خواهد شد [۹۹] و [۱۰۰]. بنابراین یکی از وظایف اصلی سیستم مدیریت شارژ، پایش وضعیت تک تک سلول های یک استک باتری و جلوگیری از افزایش ولتاژ آن ها از حد بیشینه در فرآیند شارژ و یا کاهش ولتاژ از حد کمینه مجاز در فرآیند دشارژ است. به این منظور، دو راه کار می تواند در نظر گرفته شود: ۱- قطع فرآیند شارژ و دشارژ در زمانی که ولتاژ یک سلول به مرز محدوده مجاز رسید، و ۲- یکسان سازی شارژ سلول ها بگونه ای که همواره میزان شارژ و در نتیجه سطح ولتاژ تمامی سلول ها با هم برابر باشند. در راه کار اول در انتهای فرآیند شارژ تمامی سلول ها به بیشینه ظرفیت خود نرسیده اند و بنابراین عملاً ظرفیت باتری از مقدار نامی خود کمتر است. همچنین در پایان فرآیند دشارژ نیز تعدادی از سلول ها هنوز از تمامی ظرفیت خود استفاده نکرده اند. این موارد منجر به افزایش تعداد سیکل های شارژ و دشارژ و در نتیجه کاهش طول عمر باتری می گردد. این مشکلات در راهکار دوم با یکسان سازی سطوح شارژ تمامی سلول ها مرتفع گردیده است [۱۰۳].

استراتژی های مختلفی برای یکسان سازی سطح شارژ سلول های یک باتری وجود دارد. از یک دیدگاه، روش های مختلف یکسان سازی شارژ سلول ها را می توان به دو دسته ی فعال^{۴۸} و غیر فعال^{۴۹} تقسیم نمود. تفاوت اساسی این دو گروه در آن است که در روش های یکسان سازی غیر فعال، شارژ اضافی سلول با ولتاژ بالاتر به گونه ای تلف می شود تا ولتاژ همه سلول های استک به یک سطح برسد؛ این در حالیست که در یکسان سازی های فعال، شارژ اضافی سلول مورد نظر، به سایر سلول های با ولتاژ پایین تر انتقال داده می شود [۱۰۴]. در ادامه به بررسی هر یک از این روش ها پرداخته می شود.

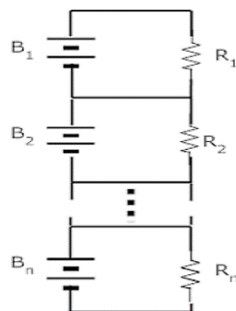
۱-۴-۱- روش های یکسان سازی شارژ غیر فعال

معمولاً عمل یکسان سازی روی سلول های استک، بیشتر در هنگام شارژ نیاز می باشد. در روش های یکسان سازی شارژ غیر فعال، یک مقاومت به هر سلول اختصاص داده می شود که به صورت موازی با آن قرار گرفته که وظیفه اتلاف شارژ اضافی سلول را بر عهده دارد. در ساده ترین سناریو، مقاومت بدون واسطه ی هیچ کلید فعالی به موازات هر سلول قرار می گیرد (شکل ۱۲-۱). در این روش مقدار جریانی که به عناصر شنت می رسد با ولتاژ سلول متناسب است؛ بنابراین هر چه ولتاژ سول افزایش یابد مقدار جریانی که به شنت فرستاده می شود بیشتر می شود که این امر منجر به کاهش اختلاف ولتاژ بین سلول های مختلف در استک می شود؛ چراکه سلول های با ولتاژ بالاتر، بیشتر از سایر سلول ها توسط المان های شنت دشارژ می شوند. یکی از معایب این روش این تلفات زیاد انرژی در عناصر شنت است. علاوه بر این، مقدار

⁴⁸ Active

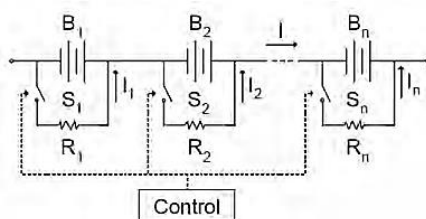
⁴⁹ Passive

جریانی که به عناصر شنت می‌رسد تنظیم شده نیست، بنابراین ولتاژ سلول‌ها هم کاملاً تنظیم نمی‌شود [۱۰۵].



شکل ۱۲-۱- مدار یکسان‌ساز غیر فعال بدون قابلیت کنترل [۱۰۵]

سناریوی دیگری که در مدارهای یکسان‌ساز شارژ غیر فعال در سلول‌های متصل به هم به صورت سری استفاده می‌شود، در شکل ۱۳-۱ قابل مشاهده است. در این روش، ولتاژ کلیه سلول‌های استک در هر لحظه به سیستم کنترل اعلام شده و سیستم کنترل فرمان وصل را به کلید متصل به سلول با بیشترین ولتاژ ارسال می‌نماید [۱۰۶].



شکل ۱۳-۱- مدار یکسان‌ساز شارژ غیرفعال با قابلیت کنترل [۱۰۶]

در این روش، با توجه به اینکه مقاومت‌های شانت پیوسته به دو سر سلول‌ها متصل نمی‌باشند، تلفات عمل یکسان‌سازی بسیار کمتر بوده و همچنین سرعت یکسان‌سازی نیز با انتخاب مقاومت شانت مناسب قابل تنظیم است. این روش، از شارژ بیش از حد سلول‌هایی که شارژ کامل شده‌اند جلوگیری می‌کند، در حالیکه سلول‌های شارژ نشده، همچنان بصورت تدریجی شارژ می‌شوند.

به این ترتیب معایب عمده‌ی روش یکسان‌سازی شارژ غیرفعال را می‌توان به صورت زیر بیان کرد:

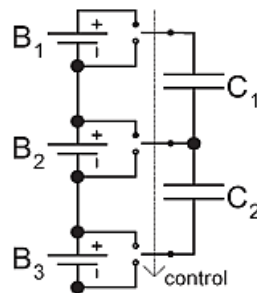
۱. عمل یکسان‌سازی تنها در هنگام شارژ باتری‌ها قابل انجام است.
۲. بخش زیادی از انرژی رسیده به استک به صورت تلفات در می‌آید و در نتیجه بازدهی این روش یکسان‌سازی نیز مناسب نمی‌باشد.
۳. در صورتی که مقدار تلفات در مقاومت‌های شنت بالا باشد، این مقاومت‌ها نیاز به خنک‌سازی دارند [۱۰۷].

۲-۴-۱- یکسان‌سازهای شارژ فعال

در این یکسان‌سازها، انرژی اضافی سلول‌ها به جای اینکه تلف شود، به وسیله‌ی عناصر فعالی نظیر خازن، میدل‌های DC به DC ایزوله و یا غیر ایزوله و یا ترکیبی از آن‌ها به سلول‌های با شارژ کمتر منتقل گشته و به این ترتیب باعث بالا رفتن بازدهی این روش‌ها خواهد شد. در قسمت‌های بعدی به بررسی رایج‌ترین طرح‌های یکسان‌سازی فعال که در خودروهای الکتریکی و الکتریکی ترکیبی به کار گرفته می‌شوند خواهیم پرداخت.

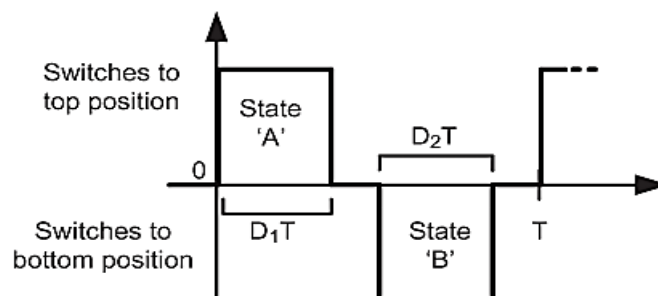
یکسان‌سازهای فعال مبتنی بر خازن‌های سوئیچ شونده

در حقیقت تمامی طرح‌های یکسان‌سازی فعال در یک مشخصه مشترکند و آن استفاده از یک یا چند عنصر فعال جهت جابه‌جایی انرژی بین سلول‌های مختلف می‌باشد. در یکی از این طرح‌ها، همانطور که در شکل ۱۴-۱ نمایش داده شده است، بین هر دو سلول موجود در استک یک خازن قرار داده می‌شود. در این طرح که به یکسان‌ساز خازنی تک‌شاخه^{۵۰} معروف می‌باشد، سوئیچ‌ها با یک فرکانس ثابت باز و بسته می‌شوند و اصلاً سطح شارژ سلول‌ها مد نظر نمی‌باشد [۱۰۸]. در حقیقت مزیت اصلی چنین سیستم یکسان‌سازی در آن است که هیچگونه المان اندازه‌گیری و یا کنترل حلقه‌بسته در آن موجود نمی‌باشد و در نتیجه پیاده‌سازی آن بسیار ساده است [۱۰۹].



شکل ۱۴-۱- مدار یکسان‌ساز خازنی تک‌شاخه [۱۰۸]

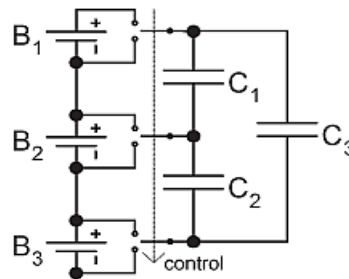
الگوریتم کلیدزنی چنین طرحی در شکل ۱۵-۱ نمایش داده شده است. در حالت "A" که حدود نیمی از دوره زمانی باز و بسته شدن کلید را در بر می‌گیرد، خازن‌ها از طریق سوئیچ به سلول بالایی متصل هستند و در حالت نشان داده شده با "B" سلول‌های پایینی به خازن‌ها متصلند. معمولاً طول دو فاصله زمانی D1 و D2 با هم مساوی در نظر گرفته می‌شود. جهت اطمینان از عملکرد صحیح مدار، یک فاصله‌ی استراحت بین این دو حالت در نظر گرفته می‌شود که طی آن، خازن‌ها به هیچ سلولی متصل نمی‌باشند. در حقیقت این زمان استراحت نسبت به دوره وصل بودن خازن‌ها به سلول‌های بالایی و پایینی بسیار کوتاه‌تر است و بعد از قرار گرفتن مدار در حالت ماندگار اعمال می‌شود تا در طول این دوره هیچ‌گونه انتقال انرژی رخ ندهد. مجموع زمان D1، D2 و دوره‌ی استراحت به عنوان دوره‌ی تناوب کلیدزنی تعریف شده و عکس آن معرفی‌کننده‌ی فرکانس سوئیچینگ می‌باشد.



⁵⁰- Single-tiered

شکل ۱۵-۱- منطق کلیدزنی یکسان ساز خازنی تک شاخه [۱۰۹]

شکل ۱۶-۱ یک یکسان ساز خازنی دو شاخه را نشان می دهد. همانطور که در شکل مشخص است، یکسان سازهای خازنی دو شاخه^{۵۱} یک مسیر اضافی را برای انتقال شارژ میان سلول های مختلف فراهم می کنند و در نتیجه تنها سلول های مجاور قابلیت تبادل شارژ آنی با یکدیگر را ندارند.

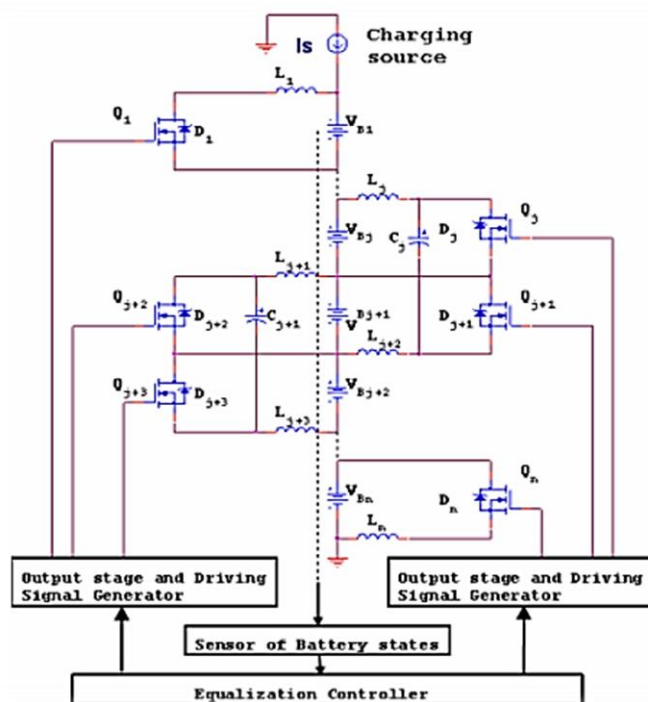


شکل ۱۶-۱- یکسان ساز خازنی دو شاخه [۱۰۹]

یکسان سازهای بر پایه مبدل های دو طرفه

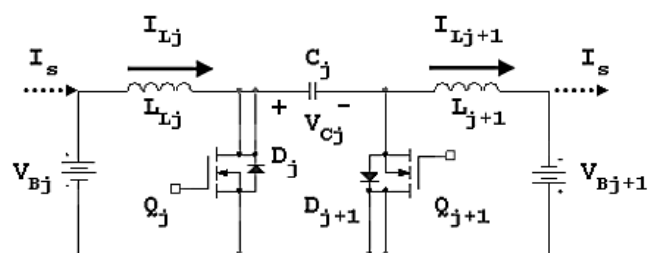
در این شیوهی متعادل سازی شارژ سلول ها، به ازای هر دو سلول سری مجاور یکدیگر از یک واحد یکسان ساز جداگانه استفاده می گردد (شکل ۱۷-۱). با توجه به طرح خاصی که در این روش مورد استفاده قرار می گیرد، به ازای n سلول سری، به $n+1$ ماژول یکسان ساز جداگانه نیاز می باشد. این ماژول ها همانطور که در شکل ۱۸-۱ نیز به نمایش درآمده است، هریک مشتمل بر دو سوئیچ ماسفت، دو اندوکتانس و یک خازن می باشند. نحوه و ترتیب کلیدزنی ها توسط یک بخش مدیریت باتری که مبتنی بر یک میکروپروسسور به عنوان مغز تصمیم گیرنده ی کل سیستم است صورت می پذیرد [۱۱۰].

⁵¹- Double-tiered



شکل ۱۷-۱- سیستم یکسان سازی مبتنی بر دایورترهای جریان دوطرفه [۱۱۰]

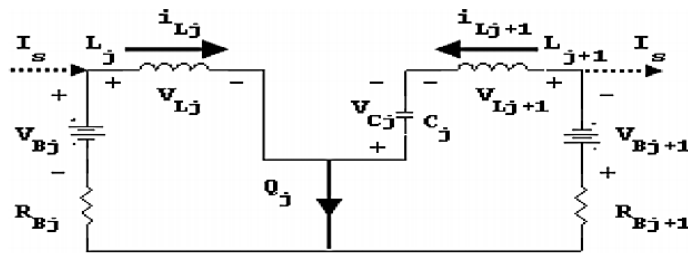
با توجه به فرمانی که از سوی سیستم مدیریت باتری برای ماسفت های هر مبدل فرستاده می شود، ولتاژ دو سلول مجاور به یکدیگر نزدیکتر می شوند. ولتاژ اولیه ی خازن C_j برابر با مجموع ولتاژ دو سلول مبدل مربوطه می باشد. برای مثال، در صورتی که سیستم کنترل مرکزی تشخیص دهد که سلول B_j محتوی شارژ بیشتری نسبت به B_{j+1} است، سیگنال PWM کنترل، سویچ Q_j را روشن و خاموش می کند و انرژی از سلول B_j به سلول B_{j+1} انتقال می یابد.



شکل ۱۸-۱- ساختار داخلی مبدل های دوطرفه [۱۱۰]

اندوکتانس های L_j و L_{j+1} به قدری بزرگ فرض می شوند که بتوان جریان گذرنده از آن ها را پیوسته فرض نمود. از طرف دیگر، خازن نیز به اندازه ی کافی کوچک فرض می گردد تا بتواند در هر پریود سوئیچینگ کاملاً دشارژ شود. T_s به عنوان دوره تناوب سوئیچ زنی مبدل تعریف می گردد. شکل ۱۹-۱ تا شکل ۲۱-۱ نشان دهنده ی مدار معادل یکسان ساز در وضعیتی که سلول B_j محتوی شارژ بیشتری نسبت به B_{j+1} است می باشد. شکل ۲۲-۱ نیز نمایانگر شکل موج ولتاژها و جریان های مختلف مدار در فاصله های زمانی مختلف می باشد.

در لحظه t_0 که سلول B_j محتوی شارژ بیشتری نسبت به B_{j+1} است، v_{Cj} قبل از اینکه Q_j روشن شود به مقدار ماکزیمم خود رسیده است. فاصله زمانی t_0 تا t_1 برابر با $D_1 T_s$ بوده و در طول این زمان، سوییچ Q_j روشن بوده و دیود D_{j+1} خاموش می‌باشد. انرژی خروجی از B_j به اندوکتانس L_j وارد شده و در آن ذخیره می‌گردد. از طرف دیگر انرژی ذخیره شده در C_j به طور کامل به سلول B_{j+1} انتقال می‌یابد و در لحظه t_1 انرژی این خازن به صفر می‌رسد.



شکل ۱۹-۱- وضعیت مداری مبدل در حالتی که Q_j روشن و D_{j+1} خاموش می‌باشد. $(V_{Bj} > V_{Bj+1})$ [۱۱۰]

معادلات دینامیکی حاکم بر مدار در این وضعیت را می‌توان به صورت ذیل بیان نمود:

$$L_j \frac{di_{Lj}}{dt} = (I_s - i_{Lj})R_{Bj} + V_{Bj}, \quad i_{Lj}(t_0) = I_0 \quad (1-58)$$

$$L_{j+1} \frac{di_{Lj+1}}{dt} = -(I_s + i_{Lj+1})R_{Bj+1} + v_{Cj} - V_{Bj+1}, \quad i_{Lj+1}(t_0) = I_0^* \quad (1-59)$$

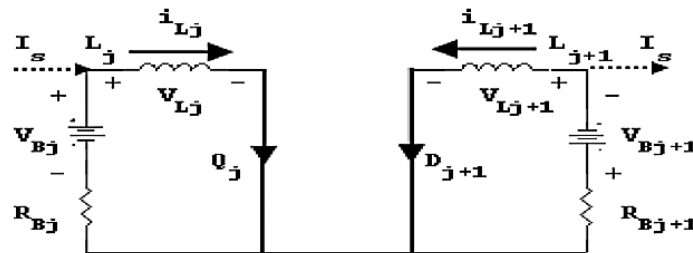
$$C_j \frac{dv_{Cj}}{dt} = i_{Lj+1}, \quad v_{Cj}(t_0) = V_{CP} \quad (1-60)$$

در فاصله زمانی t_1 تا $t_2 = DT_s$ همچنان Q_j روشن بوده و D_{j+1} نیز جریان را هدایت می‌نماید. در طول این دوره، اندوکتانس L_j همچنان توسط B_j شارژ می‌شود و انرژی بیشتری از L_{j+1} به سلول B_{j+1} منتقل می‌شود. در طول این زمان، معادلات حاکم بر متغیرهای حالت به صورت زیر می‌باشند:

$$L_j \frac{di_{Lj}}{dt} = (I_s - i_{Lj})R_{Bj} + V_{Bj}, \quad i_{Lj}(t_1) = \frac{D_1 I_p + (D - D_1) I_0}{D} \quad (1-61)$$

$$L_{j+1} \frac{di_{Lj+1}}{dt} = -(I_s + i_{Lj+1})R_{Bj+1} - V_{Bj+1}, \quad i_{Lj+1}(t_1) = \frac{D_1 I_p^* + (D - D_1) I_0^*}{D} \quad (1-62)$$

$$v_{Cj}(t_1) = 0 \quad (1-63)$$



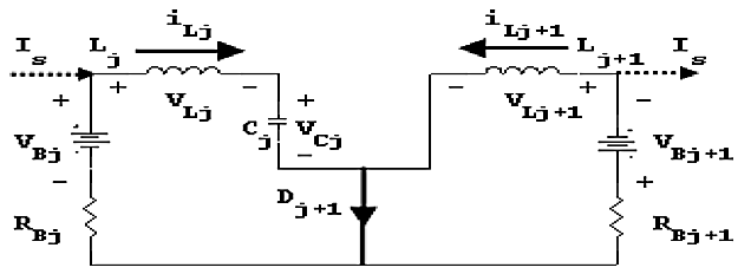
شکل ۲۰-۱- وضعیت مداری مبدل در حالتی که Q_j و D_{j+1} روشن می‌باشند. $(V_{Bj} > V_{Bj+1})$ [۱۱۰]

طی فاصله زمانی t_2 تا $t_3 = T_s$ ، سوییچ Q_j باز بوده و D_{j+1} همچنان به هدایت خود ادامه می دهد. در طول این فاصله زمانی در حالی که عمل یکسان سازی شارژ همچنان ادامه دارد، خازن C_j نیز از ولتاژ صفر شروع به شارژ مجدد می کند.

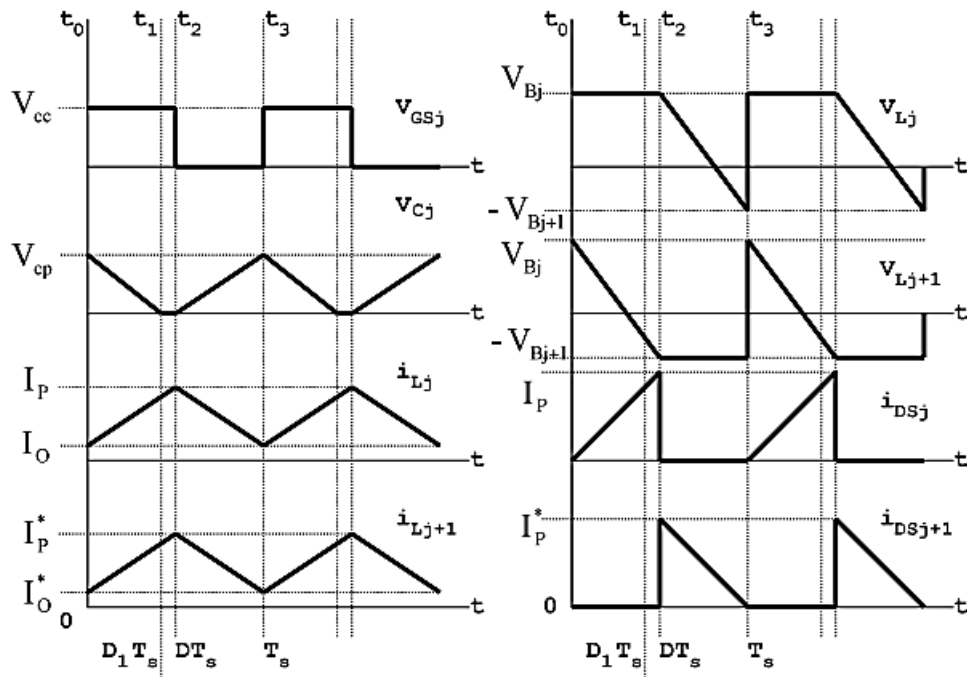
$$L_j \frac{di_{Lj}}{dt} = (I_s - i_{Lj})R_{Bj} - v_{Cj} + V_{Bj}, \quad i_{Lj}(t_2) = I_p \quad (1-64)$$

$$L_{j+1} \frac{di_{Lj+1}}{dt} = -(I_s + i_{Lj+1})R_{Bj+1} - V_{Bj+1}, \quad i_{Lj+1}(t_0) = I_0^* \quad (1-65)$$

$$C_j \frac{dv_{Cj}}{dt} = i_{Lj}, \quad v_{Cj}(t_0) = 0 \quad (1-66)$$



شکل ۲۱-۱- وضعیت مداری مبدل در حالتی که Q_j خاموش و D_{j+1} روشن می باشد. $(V_{Bj} > V_{Bj+1})$ [۱۱۰]

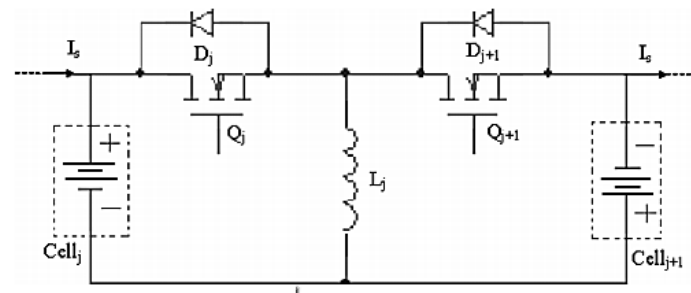


شکل ۲۲-۱- شکل موج سیگنال های مهم مدار مبدل در فواصل زمانی مختلف $(V_{Bj} > V_{Bj+1})$ [۱۱۰]

یکسان سازی با استفاده از مبدل شبه رزونانت دوطرفه

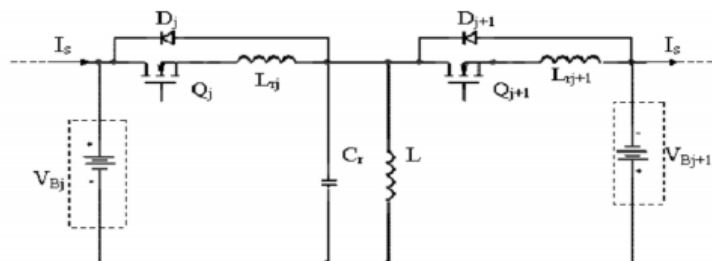
در شکل ۲۳-۱ یک مبدل باک-بوست مورد استفاده در یکسان سازی نمایش داده شده است. چنین طرح یکسان سازی دارای مزایایی از قبیل راندمان بالا، قابلیت انتقال انرژی دو طرفه و همچنین قابلیت

پیاده‌سازی مازولار می‌باشد. به سبب مشکلاتی از قبیل آلودگی الکترومغناطیسی بالا^{۵۲} و تلفات سوئیچینگ نسبتاً زیاد، جایگزین‌هایی برای این یکسان‌سازها پیشنهاد شده‌اند. مبدل شبه رزونانت با جریان صفر یکی از این مبدل‌هاست که می‌تواند با پایین آوردن تلفات کلیدزنی و کاهش EMI راندمان این یکسان‌سازها را بهبود بخشد [۱۱۱].



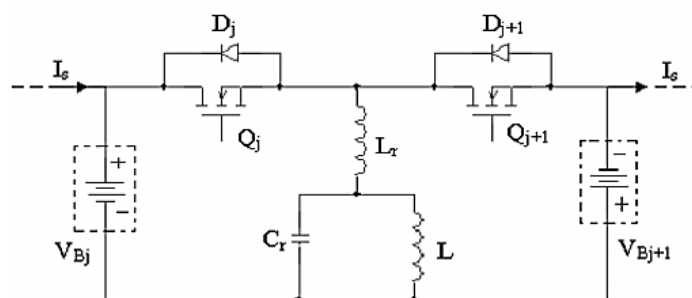
شکل ۱-۲۳ مبدل باک-بوست مورد استفاده در یکسان‌سازهای دوطرفه [۱۱۱]

جهت دستیابی به ویژگی‌های مبدل شبه رزونانت با جریان صفر، بایستی تغییراتی در مدار باک-بوست شکل ۱-۲۳ اعمال نمود. شکل ۱-۲۴ و شکل ۱-۲۵ نمایشگر دو طرح مناسب جهت اعمال چنین تغییراتی می‌باشند. مدار نشان داده شده در شکل ۱-۲۴ از دو اندوکتانس رزونانس L_{rj} و L_{rj+1} ، یک خازن رزونانس C_r ، یک اندوکتانس بزرگ ذخیره‌کننده انرژی L و دو ماسفت قدرت به همراه دیودهای مربوطه تشکیل شده است. با فرض اینکه اندوکتانس L دارای مقدار نسبتاً بزرگی است، می‌توان فرض کرد که جریان گذرنده از آن نسبتاً مقدار ثابتی می‌باشد.



شکل ۱-۲۴ مبدل باک-بوست QRZCS عادی [۱۱۱]

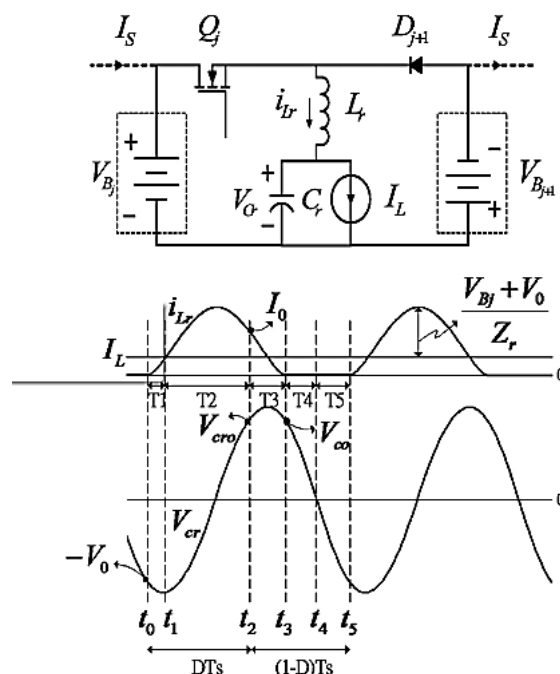
در شکل ۱-۲۵ صورت اصلاح شده‌ی مبدل QRZCS نمایش داده شده است که با استفاده از کمترین عناصر فعال و غیرفعال قادر است الگوریتم مربوط به مبدل‌های شبه رزونانت با جریان صفر (QRZCS) را پیاده‌سازی نماید.



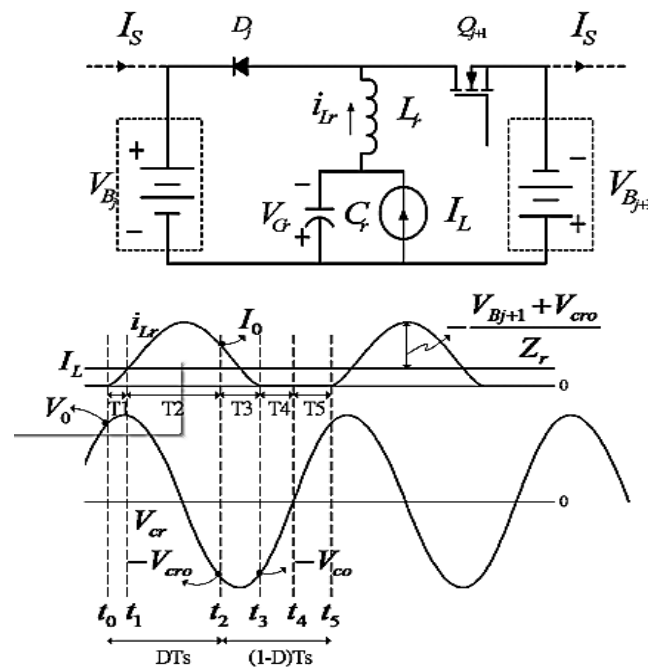
شکل ۲۵-۱- مبدل QRZCS اصلاح شده [۱۱۱]

با توجه به مقدار ولتاژ سلول‌های مجاور، جهت جریان یافتن انرژی بین دو سلول از طریق فرمان ارسالی به ماسفت‌های قدرت مشخص می‌گردد. در صورتیکه $V_{Bj} > V_{Bj+1}$ باشد، Q_j روشن شده و در زمان DT_s اندوکتانس L انرژی دریافتی از B_j را در خود ذخیره می‌کند. در ادامه و در طول دوره $(1 - DT_s)$ انرژی ذخیره شده در اندوکتانس به B_{j+1} می‌رسد. المان‌های رزونانس یعنی C_r و L_r جهت دستیابی به عملکرد شبه رزونانت با جریان صفر در مدار تعبیه شده‌اند.

وضعیت مدار و شکل موج سیگنال‌های اساسی آن در شکل ۲۶-۱ و شکل ۲۷-۱ برای دو وضعیت ولتاژی $V_{Bj} > V_{Bj+1}$ و $V_{Bj} < V_{Bj+1}$ نشان داده شده است. در ادامه جهت تحلیل مداری این طرح، لازم است چند فرض در نظر گرفته شود. اول اینکه مقدار اندوکتانس اصلی L به مقدار کافی بزرگ می‌باشد. همچنین سوییچ‌ها نیز ایده‌آل در نظر گرفته شده و عناصر راکتیو مدار تانک ایده‌آل فرض می‌گردند. با توجه به این مفروضات، جریان گذرنده از اندوکتانس L ثابت می‌ماند. جهت جریان I_L با توجه به مقادیر ولتاژ سلول‌ها تعیین می‌شود.



شکل ۲۶-۱- وضعیت مدار مبدل QRZCS در حالت $V_{Bj} > V_{Bj+1}$ [۱۱۱]



شکل ۲۷-۱- وضعیت مدار مبدل QRZCS در حالت $V_{Bj} < V_{Bj+1}$ [۱۱۱]

در ادامه مراحل مختلف عملکرد مبدل دوسویه مورد نظر با فرض $V_{Bj} > V_{Bj+1}$ بررسی می‌شود:

• مرحله نخست ($t_0 < t < t_1$):

در طول این مرحله کلید Q_j روشن بوده و اندوکتانس L از طریق مدار رزونانس L_r و C_r انرژی را در خود ذخیره می‌کند. ولتاژ اولیه خازن C_r برابر با $-V_0$ می‌باشد. معادله دینامیکی حاکم بر سیستم در طول این زمان به شرح ذیل می‌باشد.

$$L_r \frac{di_{Lr}(t)}{dt} + \frac{1}{C_r} \int (i_{Lr}(t) - I_L) dt = V_{Bj} \quad (1-67)$$

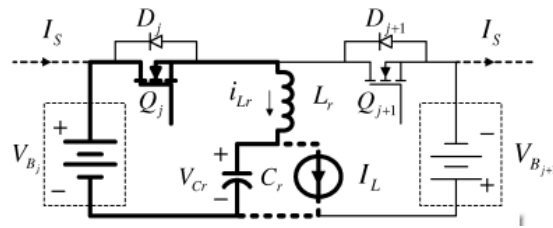
با مشتق‌گیری از دو طرف رابطه فوق، معادله درجه دوی زیر حاصل می‌گردد:

$$\frac{d^2 i_{Lr}(t)}{dt^2} + \omega_r^2 i_{Lr}(t) = \omega_r^2 I_L \quad (1-68)$$

با در نظر گرفتن شرایط اولیه $i_{Lr}(t_0) \cong 0$ و $\frac{di_{Lr}(t_0)}{dt} = \frac{V_{Bj} + V_0}{L_r}$ پاسخ معادله دیفرانسیل فوق به صورت ذیل درمی‌آید:

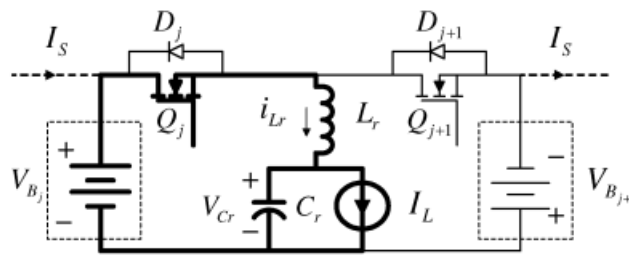
$$i_{Lr}(t) = \frac{V_{Bj} + V_0}{Z_r} \sin(\omega_r t) + (1 - \cos(\omega_r t)) I_L \quad (1-69)$$

در روابط فوق، $\omega_r = \frac{1}{\sqrt{L_r C_r}}$ فرکانس رزونانس بوده و $Z_r = \sqrt{L_r / C_r}$ امپدانس نرمالایز شده می‌باشد. در زمان t_1 جریان اندوکتانس رزونانس L_r به مقدار I_L رسیده است.

شکل ۲۸-۱- وضعیت مدار در فاصله زمانی $t_0 < t < t_1$ [۱۱۱]

• مرحله دوم: $(t_1 < t < t_2)$:

در طول این مرحله نیز کلید Q_j همچنان روشن مانده و جریان اندوکتانس به مقدار $\frac{V_{BJ}+V_0}{Z_r}$ می‌رسد. این مقدار ولتاژ نسبت به جریان DC اندوکتانس I_L یعنی بیان شده و حداکثر جریانیست که پیاده‌سازی مبدل‌های با جریان صفر را ممکن می‌سازد. جریانی که در آن این مرحله به پایان می‌رسد، یعنی I_0 دارای مقداری بین I_L و $I_L + \frac{V_{BJ}+V_0}{Z_r}$ انتخاب می‌شود.

شکل ۲۹-۱- وضعیت مدار در فاصله زمانی $t_1 < t < t_2$ [۱۱۱]

• مرحله سوم: $(t_2 < t < t_3)$:

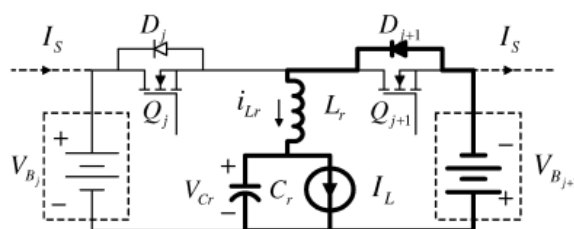
در این مرحله سوییچ Q_j قطع می‌شود و جریان اندوکتانس L_r از مسیر مشتمل بر دیود D_{j+1} جاری می‌گردد. معادله‌ی دینامیکی مدار در این وضعیت می‌تواند به صورت زیر نوشته شود:

$$L_r \frac{di_{Lr}(t)}{dt} + \frac{1}{C_r} \int (i_{Lr}(t) - I_L) dt = -V_{BJ+1} \quad (۱-۷۰)$$

با فرض اینکه $i_{Lr}(t_2) = I_0$ و $\frac{di_{Lr}(t_2)}{dt} = -\frac{(V_{BJ+1} + V_{cro})}{L_r}$ و $V_{Cr}(t_2) = V_{cro}$ پاسخ معادله دیفرانسیل فوق می‌تواند به شکل زیر بیان گردد:

$$i_{Lr}(t) = (I_0 - I_L) \cos[\omega_r(t - t_2)] - \frac{V_{BJ+1} + V_{cro}}{Z_r} \sin[\omega_r(t - t_2)] + I_L \quad (۱-۷۱)$$

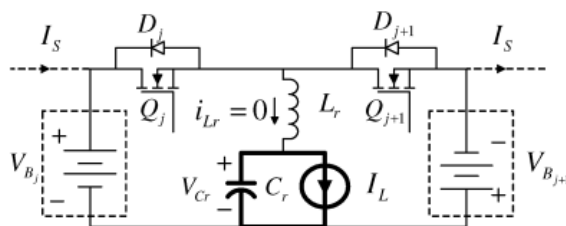
در انتهای این مرحله جریان اندوکتانس کاهش یافته و به صفر می‌رسد.



شکل ۳۰-۱ وضعیت مدار در فاصله زمانی $t_2 < t < t_3$ [۱۱۱]

• مرحله چهارم ($t_3 < t < t_4$):

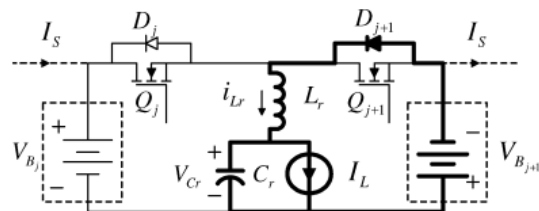
۴. در این مرحله هر دوی سوئیچ Q_j و D_{j+1} خاموش می باشند و خازن C_r در اندوکتانس L تخلیه می شود تا جاییکه ولتاژ آن به صفر برسد.



شکل ۳۱-۱ وضعیت مدار در فاصله زمانی $t_3 < t < t_4$ [۱۱۱]

• مرحله پنجم ($t_4 < t < t_5$):

در این مرحله جریان خازن به صفر رسیده و در نتیجه دیود هرزگرد D_{j+1} مجبور به هدایت می شود. حالت مداری این مرحله عیناً مشابه وضعیت مرحله سوم است و بنابراین در این زمان نیز ولتاژ روی خازن C_r به فرم سینوسی افزایش می یابد تا بار دیگر به مقدار اولیه ی خود برسد.

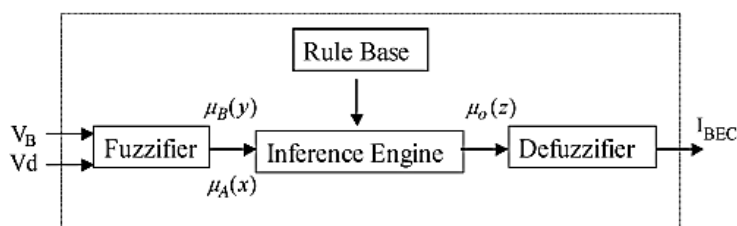


شکل ۳۲-۱ وضعیت مدار در فاصله زمانی $t_4 < t < t_5$ [۱۱۱]

بکارگیری منطق فازی در کنترل یکسان سازهای مبتنی بر مبدل های دوطرفه

این روش که نخستین بار در [۱۱۲] ارائه گردید، از همان مبدل های دوطرفه مرسوم که در بخش قبل معرفی گردید به عنوان سخت افزار استفاده می کند. در این طرح از یک استراتژی کنترلی مبتنی بر منطق فازی جهت تنظیم جریان یکسان سازی هریک از طبقه ها استفاده گردیده است. در ادامه، نحوه ی تصمیم گیری این کنترل کننده شرح داده خواهد شد.

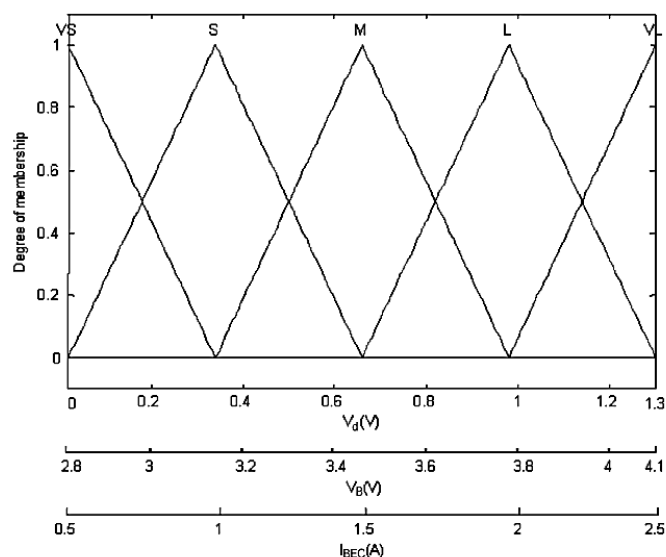
در شکل ۳۳-۱ ساختار بلوک دیاگرامی بخش کنترل نمایش داده شده است. این بخش شامل یک فازی کننده، یک ماشین استنتاج بر پایه ی قواعد از پیش تعیین شده و یک برگرداننده از فضای فازی می باشد.



شکل ۳۳-۱- بلوک دیاگرام سیستم کنترل مبتنی بر منطق فازی [۱۱۲]

در شکل فوق ورودی‌های μ_A و μ_B و همچنین خروجی μ_o به ترتیب توابع عضویت متناظر با اختلاف ولتاژ دو سلول مجاور، مقدار ولتاژ سلول و جریان یکسان‌سازی می‌باشند. قواعد به کار گرفته شده در بخش استنتاج فازی، همگی بر اساس دانش و تجربه‌ی مربوط به سیستم یکسان‌سازی تولید و گردآوری شده‌اند. مزیت این روش کنترلی در آن است که نیازی به دانستن مدل دقیق و پیچیده باتری نیست. سیستم کنترل فازی دارای دو ورودی V_B و V_d به عنوان اختلاف ولتاژ دو سلول و ولتاژ سلول می‌باشد. این ورودی‌ها توسط بخش فازی کننده، به مجموعه‌های زبانی فازی تبدیل می‌گردند و با توجه به این ورودی‌ها و قواعد فازی ذخیره شده، خروجی‌های کنترلی زبانی در بخش استنتاج فازی تولید می‌گردند. بخش برگرداننده از فازی وظیفه تبدیل این خروجی‌های زبانی به مقادیر عددی I_{BEC} را بر عهده دارد. این مقادیر به عنوان جریان مطلوب جهت ادامه عمل یکسان‌سازی تعیین می‌شوند.

بنابراین جریان یکسان‌سازی با توجه به ولتاژهای اندازه‌گیری شده V_B و V_d تعیین می‌گردد. شکل ۳۴-۱ نمایش دهنده توابع عضویت برای این استراتژی یکسان‌سازی می‌باشد. قسمت بالایی در این شکل مربوط به متغیر فازی μ_A با توجه به اختلاف ولتاژ دو سلول مجاور یعنی V_d می‌باشد. این متغیر با استفاده از متغیر زبانی یعنی V_L (بسیار بزرگ)، L (بزرگ)، M (متوسط)، S (کوچک) و V_S (بسیار کوچک) و به شکل مثلثی توصیف شده است. متغیر فازی μ_B با توجه به اختلاف ولتاژ دو سلول مجاور یعنی V_B نیز توسط همین پنج تابع مثلثی شکل تعریف شده و در قسمت میانی شکل مشخص شده است. در قسمت پایینی این شکل نیز توابع عضویت مربوط به خروجی فازی μ_o با توجه به مقادیر I_{BEC} مشخص شده است. مجموعه قواعد فازی در جدول ۳-۱ خلاصه شده‌اند.



شکل ۳۴-۱- توابع عضویت مربوط به استراتژی فازی برای تنظیم جریان یکسان سازی [۱۱۲]

جدول ۳-۱- قواعد فازی استفاده شده در ماشین استنتاج برای متغیرهای زبانی [۱۱۲]

		V_d				
V_B	Output	VS	S	M	L	VL
	VS	VS	M	L	VL	VL
	S	VS	M	L	VL	VL
	M	VS	M	L	VL	VL
	L	VS	S	M	L	VL
	VL	VS	S	M	L	VL

الف) مقادیر صحیح ω_{ij} برای λ امین ورودی x و λ امین ورودی y به صورت زیر بدست می آید:

$$\omega_{ij} = \min\{\mu_{Ai}(x), \mu_{Bj}(y)\}, \text{ for } i = 1, 2, \dots, 5; j = 1, 2, \dots, 5. \quad (1-72)$$

ب) تعیین خروجی فازی $\mu_{oij}(z)$ با توجه به ω_{ij} و k امین خروجی تابع عضویت که به صورت زیر انجام می پذیرد:

$$\mu_{oij}(z) = \min\{\omega_{ij}, \mu_{ok}(z)\} \text{ for } k = 1, 2, \dots, 5. \quad (1-73)$$

پ) محاسبه متغیر فازی خروجی برای خروجی z یعنی $\mu_{out}(z)$:

$$\mu_{out}(z) = \max\{\mu_{o11}(z), \mu_{o12}(z), \dots, \mu_{o54}(z), \mu_{o55}(z)\} \quad (1-74)$$

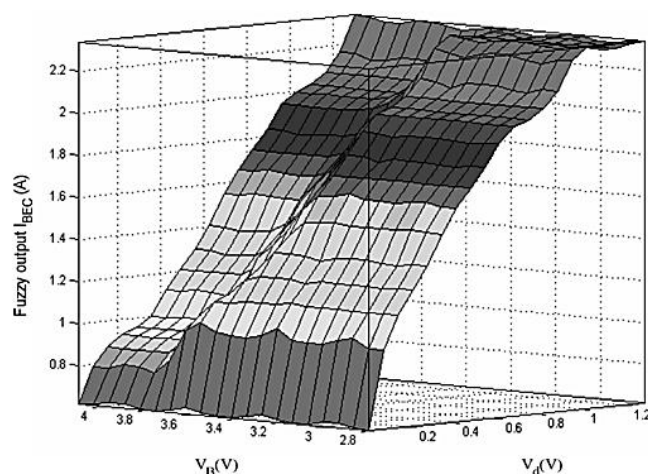
ت) محاسبه جریان یکسان سازی مطلوب با استفاده از روش مرکز جرم:

$$I_{BEC} = \frac{\sum_{i=1}^m \mu_{out}(z_i) \cdot z_i}{\sum_{i=1}^m \mu_{out}(z_i)} \quad (1-75)$$

ث) در پایان مقادیر بدست آمده برای I_{BEC} از طریق رابطه زیر، دوره تناوب سوئیچینگ PWM را تعیین می نمایند.

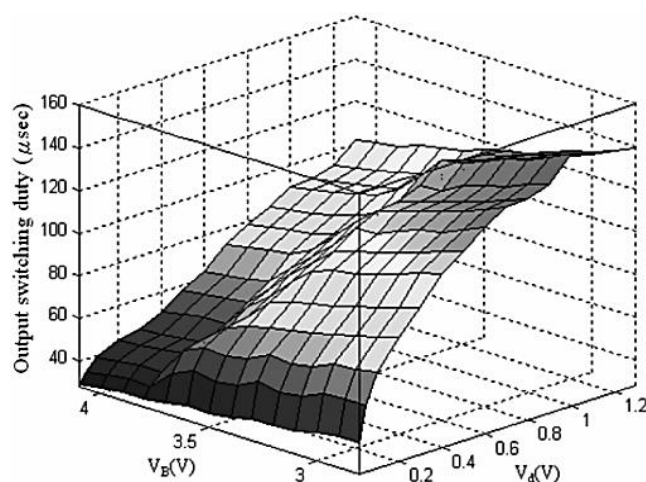
$$T_s = 2I_{BEC} / \left(\frac{V_{B1}}{L_1} D^2 + \frac{V_{C1} - V_{B1}}{L_1} (1 - D)^2 \right) \quad (1-76)$$

شکل ۳۵-۱- نمایشگر جریان یکسان سازی خروجی کنترل کننده فازی بر حسب V_d و V_B می باشد. همانطور که در این شکل مشاهده می شود، هرچه ولتاژ سلول و اختلاف ولتاژ در یک مازول یکسان ساز بیشتر باشند، جریان مطلوب درخواستی توسط کنترل کننده فازی افزایش می یابد.



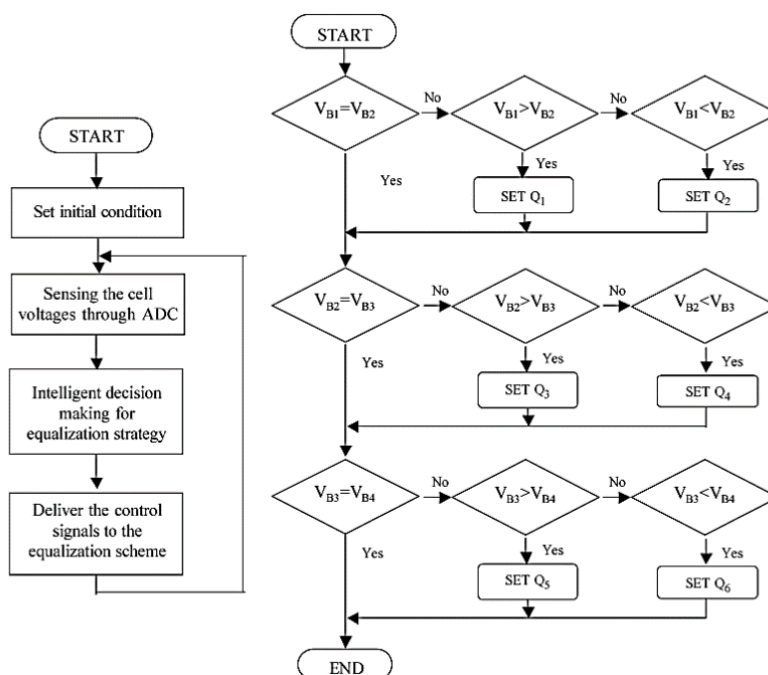
شکل ۳۵-۱- مقادیر جریان I_{BEC} تولیدی توسط کنترل کننده فازی به ازای V_B و V_d مختلف [۱۱۲]

در شکل ۳۶-۱ نیز رابطه ی درصد روشن بودن کلید و ولتاژهای V_B و V_d مختلف نشان داده شده است.



شکل ۳۶-۱- رابطه درصد روشن بودن کلید و ولتاژهای V_B و V_d مختلف [۱۱۲]

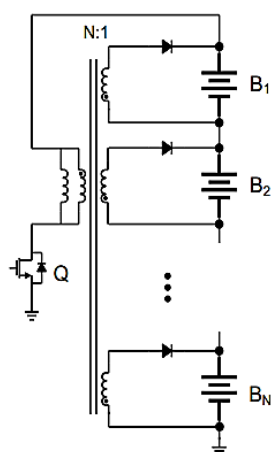
البته سیستم مدیریت باتری علاوه بر تعیین مقدار جریان یکسان سازی و مدت زمان روشن بودن کلید، نیاز به الگوریتمی دارد که بر اساس آن بتواند کلید مورد نظر را روشن نموده و فرامین بعدی را به آن ارسال نماید. شکل ۳۷-۱ بیان کننده ی چنین الگوریتمی برای استک شامل باتری های سری با هم می باشد.



شکل ۳۷-۱- فلوچارت نحوه تولید فرمان روشن و یا خاموش بودن کلیدهای مختلف [۱۱۲]

یکسان‌سازهای شارژ فعال با مبدل مبتنی بر ترانسفرمر ایزوله

بیشترین انواع یکسان‌سازهای شارژ فعال پیشنهادی برای خودروهای الکتریکی و هایبرید الکتریکی در این دسته قرار می‌گیرند [۱۱۳]. به صورت کلی این یکسان‌سازها بر دو نوع هستند. در نوع اول، کنترل خاصی بر روی تک تک سلول‌ها اعمال نمی‌شود (شکل ۳۸-۱). سمت اولیه به باس باتری متصل شده و هر کدام از سیم‌پیچ‌های ثانویه به یک سلول یا یک گروه از سلول‌ها متصل می‌شود. بنابراین تعداد سیم‌پیچ‌های ثانویه مساوی تعداد سلول‌های باتری خواهد بود.



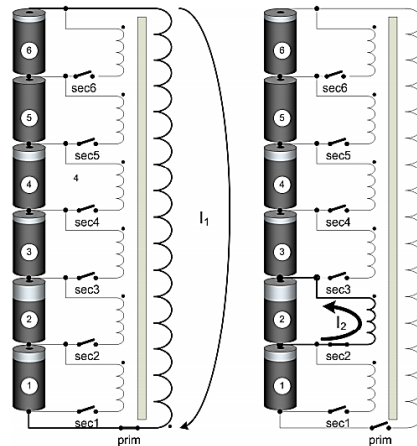
شکل ۳۸-۱- یکسان سازی شارژ با استفاده از مبدل DC-DC فلالی بک مرکزی [۱۱۳]

در این روش ولتاژ ماژول‌های باتری توسط انتقال انرژی از باس باتری به ضعیف‌ترین سلول‌ها در استک، دقیقاً تنظیم می‌شود. در حالت ایده‌آل مبدل تا وقتی که یک ماژول ضعیف پیدا شود غیرفعال است.

وقتی سوئیچ بسته شد انرژی در اندوکتانس مغناطیسی ترانسفورمر ذخیره شده و بعد توسط سوئیچ بسته شده به مازول با ولتاژ پایین انتقال می یابد.

همه کویلها دور یک هسته پیچیده شده اند؛ بنابراین همه ولتاژها یکسان خواهند بود. در حالت ایده آل بیشترین قسمت انرژی ذخیره شده بدون هر گونه کنترل اضافی به مازول با کمترین ولتاژ فرستاده می شود؛ ولی در واقعیت این طرح حساسیت بالایی به اندوکتانس نشی بین سیم پیچی های ثانویه دارد.

روش دیگری که از مبدل یکسان سازی مرکزی استفاده می کند مبدل پیش سو با ترانسفورمر چند سیم پیچه است [۱۱۴]. قطعه ای اصلی این یکسان سازی ترانسفورمر می باشد. نقش اصلی ترانسفورمر در این یکسان سازی انتقال انرژی بین سلول های مختلف استک می باشد. در حقیقت یکسان سازی های مبتنی بر مبدل های ایزوله، می توانند در دو وضعیت کاری عمل کنند. اگر استک در حین دشارژ بوده و ولتاژ یکی از باتری ها به منطقه خطر نزدیک باشد، یکسان سازی از بالا^{۵۳} اعمال می شود. در این حالت، همانطور که در شکل ۳۹-۱ مشاهده می شود ابتدا کلید اولیه ترانس بسته است و انرژی در سیم پیچ مربوطه ذخیره می گردد؛ در مرحله بعدی کلید باز شده و انرژی ذخیره شده در سلول با کمترین شارژ تخلیه می گردد.



شکل ۳۹-۱- نمایش نحوه عملکرد پیاده سازی یکسان سازی از بالا [۱۱۴]

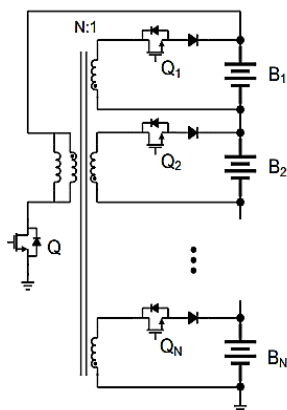
در حالتی که استک در حال شارژ بوده و یکی از سلول ها در معرض ورود به محدوده غیر مجاز ولتاژ است، یکسان سازی از پایین^{۵۴} صورت می پذیرد. در این یکسان سازی ها، شارژ ذخیره شده در سلول با بیشترین ولتاژ از طریق الکترومغناطیسی به سلول های با ولتاژ کمتر انتقال می یابد. عنصر اصلی این مبدل های فلای بک^{۵۵} یک هسته با دو و یا چند سیم پیچ است. هسته ی ترانس بایستی از جنس فریت^{۵۶} انتخاب گردد تا تلفات هیستریزس و جریان گردابی نداشته باشیم. شکل ۴۰-۱ و شکل ۴۱-۱ نمایش دهنده ی دو طرح مختلف از این یکسان سازی می باشند. در طرح شکل ۴۰-۱، به هر سلول یک سیم پیچ از هسته ی ترانس اختصاص داده شده است؛ در حالیکه طرح نمایش داده شده در شکل ۴۱-۱، یک سیم پیچ را به سلول های مختلف اختصاص می دهد. البته طرح دوم نیازمند کلیدهای اضافی می باشد.

53-Top Balancing

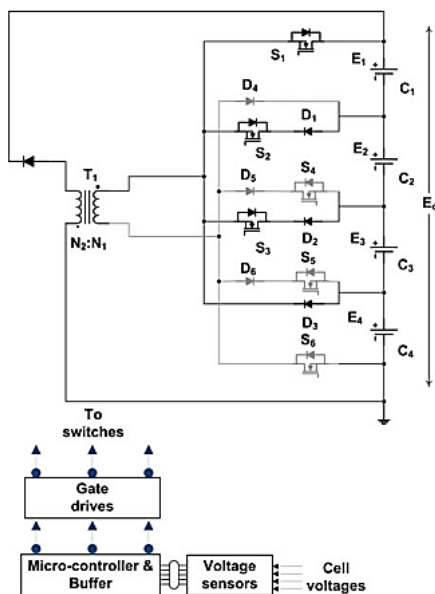
54-Bottom Balancing

55- Flyback

56- Ferrit



شکل ۴۰-۱- یکسان ساز با مبدل فلای بک مرکزی با سیم پیچ‌های ثانویه متعدد [۱۱۳]

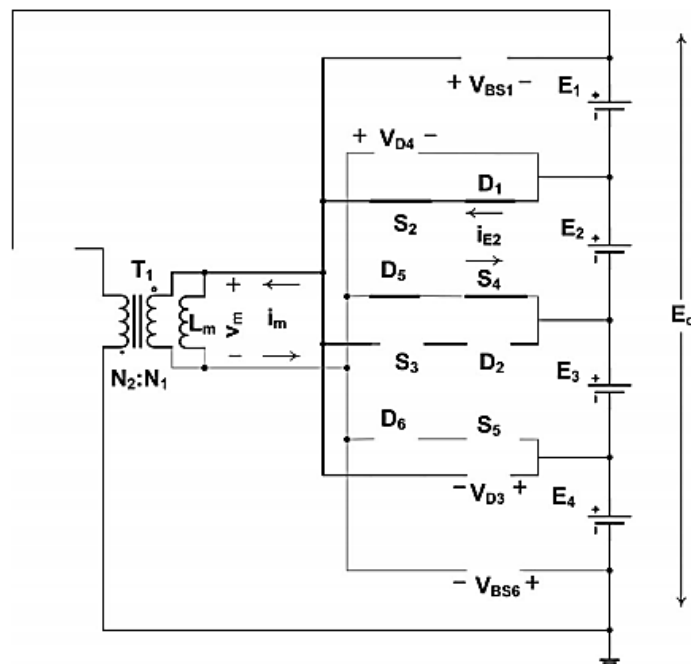


شکل ۴۱-۱- یکسان ساز با مبدل فلای بک مرکزی با یک سیم پیچ ثانویه [۱۱۵]

هر دو مدار از لحاظ عملکردی مشابه یکدیگر هستند در دومی، از تعداد بیشتری سویچ استفاده شده است که می‌تواند هزینه‌ی مدار را بالاتر ببرد. در ادامه به بررسی حالات مختلف عملکردی مدار نشان داده شده در شکل ۴-۳۲ می‌پردازیم.

الف- مرحله‌ی اول (t_0 تا t_1)

نخست سیستم کنترل، تمام ولتاژهای اندازه‌گیری شده را بررسی کرده و سلول با بیشترین ولتاژ را انتخاب می‌نماید. در صورتیکه سلول انتخاب شده، E_2 باشد، کلیدهای S_2 و S_4 بسته شده و حالت مدار به صورت نشان داده شده در شکل ۴۲-۱ در می‌آید.



شکل ۴۲-۱- وضعیت مدار یکسان‌ساز در حالتی که سلول دوم در حال دشارژ شدن است [۱۱۶]

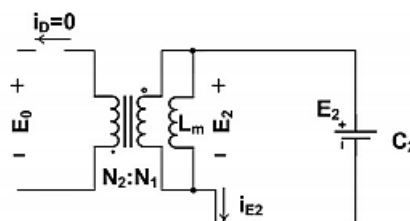
در این دوره زمانی تغییرات جریان گذرنده از سلول اول به صورت زیر صورت خواهد پذیرفت:

$$\frac{d}{dt} i_{E1} = \frac{E_1}{L_m} \quad (1-77)$$

در نتیجه در پایان این دوره، خواهیم داشت:

$$i_{E1}(t_1) = \frac{E_1}{L_m} (t_1 - t_0) \quad (1-78)$$

در مبدل‌های فلای‌بک معمولاً از روش کنترل جریان بیشینه^{۵۷} استفاده می‌کنند. در این روش، جریان گذرنده از سیم‌پیچ اندازه گرفته می‌شود و با رسیدن این جریان به یک حد بیشینه‌ای از پیش تعریف شده، کلید مربوطه جریان را قطع می‌کند. شکل ۴۳-۱ مدل دقیق یک ترانس را که در مبدل فلای‌بک به کار گرفته شده را نمایش می‌دهد.

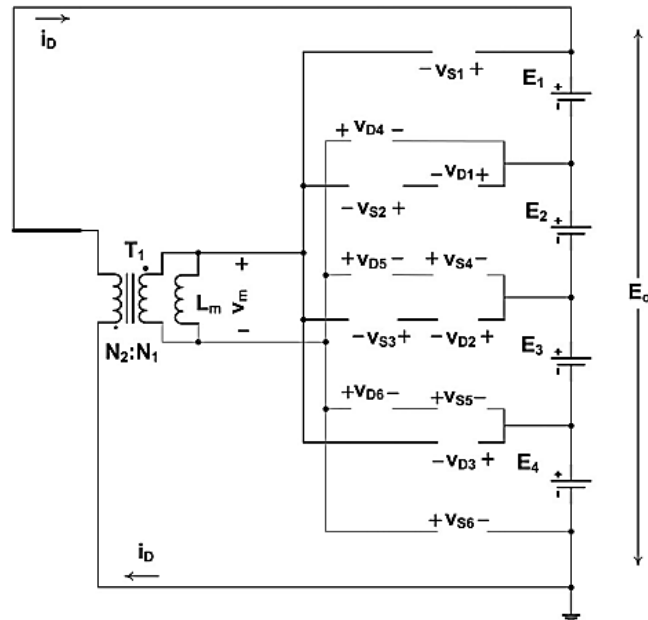


شکل ۴۳-۱- مدار ساده شده یکسان‌ساز با مبدل فلای‌بک [۱۱۶]

با توجه به اینکه در این فاصله زمانی ثانویه‌ی ترانس قطع می‌باشد، بنابراین تمام انرژی خارج شده از سلول E2 در اندوکتانس مغناطیس‌کنندگی ترانس ذخیره شده است.

ب- مرحله ی دوم (t_1 تا t_2)

پس از اینکه جریان اولیه ی ترانس به مقدار بیشینه (I_{peak}) رسید، فرمان قطع به سلول S2 و S4 صادر می شود و در نتیجه انرژی ذخیره شده در اندوکتانس مغناطیس کنندگی (L_m) به ثانویه ترانس منتقل شده و وضعیت مدار به صورت نمایش داده شده در شکل ۴۴-۱ در خواهد آمد.



شکل ۴۴-۱- وضعیت مدار یکسان سازی با مبدل فلای بک مرکزی در حالتی که دیود ثانویه هدایت می کند [۱۱۶].

در این فاصله ی زمانی، تغییرات جریان گذرنده از کل استک به صورت زیر بیان می گردد:

$$i_D(t) = i_{pack}(t) = \frac{1}{n} \left\{ i_{peak} - \frac{v_{pack}}{nL_m} (t - t_1) \right\} \quad (1-79)$$

این مرحله تا زمانی ادامه می یابد که جریان برقرار شده در ثانویه ترانس به صفر برسد. مبدل های DC به DC را می توان در دو حالت پیوسته^{۵۸} و یا گسسته^{۵۹} جریانی مورد استفاده قرار داد. در حقیقت به سبب پیچیدگی کنترل پیوسته جریانی این مبدل ها، از آن ها معمولاً در مد گسسته جریانی استفاده می گردد.

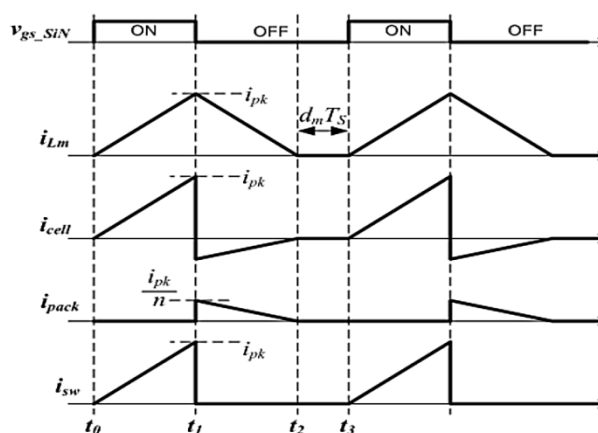
پ- مرحله سوم (t_2 تا t_3)

همانطور که در قسمت قبل بیان گردید کلید S1 تا زمانی که جریان ثانویه به صفر نرسیده باشد وصل نمی گردد. این مرحله از لحظه ای که جریان به صفر رسیده است شروع می شود و تا پایان آن هیچ یک از کلیدها هدایت نخواهند نمود.

شکل ۴۵-۱ جریان ها و ولتاژ کلیدهای مختلف را طی مراحل سه گانه ی شرح داده شده در فوق نمایش می دهد.

⁵⁸- Continious Current Mode(CCM)

⁵⁹- Discrete Current Mode(DCM)



شکل ۴۵-۱- وضعیت ولتاژها و جریان ها مختلف مدار یکسان ساز طی مراحل مختلف [۱۱۶]

در شکل فوق، v_{gs-SiN} ولتاژ بین گیت و سورس کلید متصل به سلول با بیشترین ولتاژ بوده و i_{Lm} ، i_{cell} ، i_{pack} و i_{sw} به ترتیب نشان دهنده ی جریان گذرنده از اندوکتانس مغناطیس کننده ی، جریان سلول متصل به کلید، جریان یکسان سازی کل استک و جریان گذرنده از کلید می باشند.

۵-۱- جمع بندی

همان طور که اشاره شد سیستم های مدیریت باتری جزء کلیدی وسایل نقلیه الکتریکی ترکیبی می باشند که وظیفه تأمین سلامت باتری و افزایش رادمان آن را برعهده دارند.

در این فصل در ابتدا کلیاتی در مورد باتری و تاریخچه آن عنوان گردید. انواع تکنولوژی های باتری و خواص الکتروشیمیایی هر یک معرفی و با یکدیگر مقایسه شد و باتری های لیتیوم-یون به عنوان مبنا انتخاب گردید. این باتری ها به دلیل عملکرد خوب، چگالی انرژی و توان مخصوص بالا و قابلیت اعتماد بالایشان به طور گسترده در قطعات الکترونیکی قابل حمل از جمله موبایل، دوربین های دیجیتال و لپ تاپ ها مورد استفاده قرار می گیرند. همچنین محدوده تغییرات SOC این نوع باتری ها حدود ۱۰٪ تا ۱۰۰٪ است.

در ادامه، انواع مدل های باتری مورد بررسی قرار گرفت. این مدل ها به چهار دسته مدل های ریاضی (تجربی)، مدل های الکتروشیمیایی، مدل های مدار الکتریکی و مدل های انتزاعی دسته بندی شدند. بعضی از آن ها مانند مدل الکتروشیمیایی، باتری را هم در سطح ماکروسکوپی و هم در سطح میکروسکوپی توصیف می کنند، اما بعضی دیگر فقط رفتار ورودی-خروجی باتری را مد نظر قرار می دهند. از بین مدل های مطرح شده، مدل الکتروشیمیایی دارای بیشترین دقت و مدل های ریاضی دارای دقت کمی می باشند. مدل الکتریکی نسبت به مدل الکتروشیمیایی دارای دقت کمتر و نسبت به مدل ریاضی دارای دقت بالاتری است. این مدل ها برای یک مهندس برق ساده تر، محسوس تر و پرکاربردتر هستند، چون می توان از این مدل ها برای شبیه سازی همراه با سایر مدارها و سیستم های الکتریکی استفاده کرد. برخلاف مدل های دیگر، مدل های انتزاعی بر روش های صرفاً ریاضی برای توصیف رفتار باتری مبتنی می باشد. از این مدل ها می توان به مدل های مبتنی بر هوش مصنوعی، مدل های آماری، مدل مبتنی بر SVM و مدل های مبتنی بر روش استنتاج نروفازی اشاره کرد. در این پروژه، از مدل الکتریکی برای اهداف تخمین استفاده خواهد شد.

در بخش بعد، به معرفی انواع روش های ارائه شده در مقالات برای تخمین SOC پرداخته شد. روش های ارائه شده برای تخمین SOC به طور کلی به دو دسته روش های مستقیم و غیرمستقیم دسته بندی شده اند. در روش های غیرمستقیم، SOC از روی برخی خواص فیزیکی باتری مانند چگالی اسید یا پالس های گالوانواستاتیک کاتدی تخمین زده می شود. از روش های غیرمستقیم می توان به اندازه گیری ولتاژ مدار باز باتری، طیف نمایی امپدانس و تخمین ولتاژ نیروی الکتروحرکتی اشاره کرد. روش های مستقیم تخمین SOC نیز شامل روش شمارش آمپرساعت (انتگرال گیری جریان)، شبکه های عصبی مصنوعی، فیلتر کالمن و ... می باشند. دو روش معروف ارائه شده برای تخمین حالت باتری، روش فیلتر کالمن توسعه یافته و رؤیتگر مد لغزشی می باشد. فیلتر کالمن برای تخمین SOC نیاز به یک مدل فضای حالت باتری دارد و در BMS با موفقیت مورد استفاده قرار گرفته است. رؤیتگر مد لغزشی (SMO)، یکی دیگر از رؤیتگرهای مبتنی بر مدل است که نسبتاً دقیق بوده و نسبت به عدم قطعیت در پارامترهای مدل و اغتشاش مقاوم است.

روش های مختلف یکسان سازی شارژ سلول های سری موجود در یک استک، در ادامه مورد بررسی قرار گرفتند. روش های متعادل سازی ولتاژ به دو دسته ی فعال و غیر فعال تقسیم شدند. یکسان سازی غیر فعال، شارژ اضافی سلول های با ولتاژ بالاتر را توسط یک شاخه ی مقاومتی تلف می کنند تا ولتاژ سلول های استک به یک سطح برسد. در روش های فعال یکسان سازی شارژ، به کمک یکی از انواع مبدل های مبتنی بر عناصر بدون تلف نظیر اندوکتانس، خازن و سایر مدارات الکترونیکی مربوطه، شارژ اضافی یک سلول به سلول دیگر و یا به کل استک منتقل می گردد. انواع مبدل های DC به DC دوطرفه و یکسان سازی خازنی تک شاخه و دو شاخه و همچنین یکسان سازی مبتنی بر مبدل های فلای بک، مورد بررسی قرار گرفتند. در پایان، یکسان سازی مبتنی بر مبدل های فلای بک به عنوان مناسبترین گزینه جهت اجرای طرح های کنترلی در نظر گرفته شدند.

۶-۱- مراجع

- [1] M. Chen, and G. A. Rincon-Mora, "Accurate electrical battery model capable of predicting runtime and I-V performance," *IEEE Trans. Energy Convers.*, vol. 21, no. 2, pp. 504-511, Jun. 2006.
- [2] N. A. Chaturvedi, R. Klein, J. Christensen, J. Ahmed, and A. Kojic, "Algorithms for Advanced Battery-Management Systems: Modeling, estimation, and control challenges for lithium-ion battery," *IEEE Control Syst. Mag.*, vol. 30, no. 3, pp. 49-68, Jun. 2010.
- [۳] امین رضایی پیش‌رباط، تخمین پارامترهای شارژ، تخلیه و وضعیت سلامت باتری با استفاده از روش‌های شناسایی و تخمین غیرخطی، پایان‌نامه کارشناسی ارشد، دانشگاه تهران، دانشکده مهندسی برق و کامپیوتر، شهریور ۱۳۸۶.
- [4] R. J. Grube, Automotive Battery State-of-Health Monitoring Methods, M.Sc. Thesis, Wright State University, Dec. 2008.
- [5] C. Alaoui and Z. M. Salameh, "Experiments in Fast Charging Lead Acid Electric Vehicle Batteries," *IEEE Trans. Veh. Technol.*, vol. 5, pp. 3326-3331, 2003.
- [6] M. Ehsani, Y. GAO, S. E. Gay, and A. Emadi, Modern Electric, Hybrid Electric, and Fuel Cell Vehicles, New York: CRC Press LLT, 2005.
- [7] K. M. Tsang, L. Sun, and W. L. Chan, "Identification and modeling of Lithium ion battery," *Energy Convers. Manage.*, vol. 51, pp. 2857-2862, Jun. 2010.
- [8] B. S. Bhangu, P. Bentley, D. A. Stone, and C. M. Bingham, "Nonlinear Observers for Predicting State-of-Charge and State-of-Health of Lead-Acid Batteries for Hybrid-Electric Vehicles," *IEEE Trans. Veh. Technol.*, vol. 54, no. 3, pp. 783-794, May 2005.
- [9] D. W. Dennis, V. S. Battaglia, and A. Belanger, "Electrochemical modeling of lithium polymer batteries," *J. Power Sources*, vol. 110, no. 2, pp. 310-320, Aug. 2002.
- [10] R. C. Kroeze, and P. T. Krein, "Electrical Battery Model for Use in Dynamic Electric Vehicle Simulations," *Power Electron. Specialists Conf.*, pp. 1336-1342, 2008.
- [11] K. A. Smith, C. D. Rahn, C.-Y. Wang, "Control oriented 1D electrochemical model of lithium ion battery," *Energy Convers. Manage.*, vol. 48, no. 9, pp. 2565-2578, Jun. 2007.
- [12] R. Rynkiewicz, "Discharge and charge modeling of lead acid batteries," in *Proc. Appl. Power Electron. Conf. Expo.*, vol. 2, pp. 707-710, 1999.
- [13] M. Pedram, Qing Wu, "Design considerations for battery-powered electronics," in *Proc. 36th ACM/IEEE Design Automation Conf.*, pp. 861-866, Jun. 1999.
- [14] D. Rakhmatov, S. Vrudhula, and D. A. Wallach, "A model for battery lifetime analysis for organizing applications on a pocket computer," *IEEE Trans. VLSI Syst.*, vol. 11, no. 6, pp. 1019-1030, Dec. 2003.
- [15] K.C. Syracuse, W.D.K. Clark, "A statistical approach to domain performance modeling for oxyhalide primary lithium batteries," *Battery Conf. Applications and Advances*, pp. 163-170, Jan. 1997.
- [16] K. A. Smith, C. D. Rahn, and C.-Y. Wang, "Model Order Reduction of 1D Diffusion Systems Via Residue Grouping," *ASME Trans.*, vol. 130, Jan. 2008.
- [17] M. W. Verbrugge, and P. Liu, "Electrochemical characterization of high-power lithium ion batteries using triangular voltage and current excitation sources," *J. Power Sources*, vol. 174, pp. 2-8, Mar. 2007.
- [18] Z. M. Salameh, M. A. Casacca, and W. A. Lynch, "A mathematical model for lead-acid batteries," *IEEE Trans. Energy Convers.*, vol. 7, no. 1, Mar. 1992.

- [19] M. Valvo, F. E. Wicks, D. Robertson, and S. Rudin, "Development and application of an improved equivalent circuit model of a lead acid battery," in Proc. Energy Convers. Eng. Conf., vol. 2, pp. 1159–1163, Aug. 1996.
- [20] M. Ceraolo, "New dynamical models of lead-acid batteries," IEEE Trans. Power Syst., vol. 15, no. 4, pp. 1184–1190, Nov. 2000.
- [21] S. Barsali and M. Ceraolo, "Dynamical models of lead-acid batteries: Implementation issues," IEEE Trans. Energy Convers., vol. 17, no. 1, pp. 16–23, Mar. 2002.
- [22] L. Gao, S. Liu, and R. A. Dougal, "Dynamic lithium-ion battery model for system simulation," IEEE Trans. Compon. Packag. Technol., vol. 25, no. 3, pp. 495–505, Sep. 2002.
- [23] B. Schweighofer, K. M. Raab, and G. Brasseur, "Modeling of high power automotive batteries by the use of an automated test system," IEEE Trans. Instrum. Meas., vol. 52, no. 4, pp. 1087–1091, Aug. 2003.
- [24] M. C. Glass, "Battery electrochemical nonlinear/dynamic SPICE model," in Proc. Energy Convers. Eng. Conf., vol. 1, pp. 292–297, 1996.
- [25] Y. Hu, and S. Yurkovich, "Battery State of Charge Estimation in Automotive Applications using LPV Techniques," American Control Conf., Jul. 2010.
- [26] H. Dai, Z. Sun, and X. Wei, "Online SOC Estimation of High-power Lithium-ion Batteries Used on HEV's," IEEE Int. Conf. Veh. Electron. Safety, pp. 342–347, 2006.
- [27] L. Wang, Li. Wang, and C. Liao, "Research on Improved EKF Algorithm Applied on Estimate EV Battery SOC," IEEE Conf. Power and Energy Engineering Conference (APPEEC), pp. 1–4, 2010.
- [28] K.M. Tsang, W.L. Chan, Y. K. Wong, and L. Sun, "Lithium-ion Battery Models for Computer Simulation," in Proc. IEEE Int. Conf. Automation and Logistics (ICAL), pp. 98–102, Aug. 2010.
- [29] C.-F. Chiasserini, R.R. Rao, "Energy efficient battery management," IEEE J. Selected Areas in Communications, vol.19, no.7, pp.1235–1245, Jul. 2001.
- [30] Junping Wang, Quanshi Chen, Binggang Cao, "Support vector machine based battery model for electric vehicles," Energy Convers. Manage., vol.47, no.7-8, pp.858–864, May 2006.
- [31] C.C. O'Gorman, D. Ingersoll, R.G. Jungst, T.L. Paez, "Artificial neural network simulation of battery performance," in Proc. 31th Hawaii Int. Conf. System Sciences, vol. 5, pp.115–121, Jan. 1998.
- [32] K. T. Chau, K. C. Wu and C. C. Chan, "A new battery capacity indicator for lithium-ion battery powered electric vehicles using adaptive neurofuzzy inference system," Energy Convers. Manage., vol.45, no.11-12, pp.1681–1692, Jul. 2004.
- [33] Mehmet Ugras Cuma, Tahsin Koroglu, "A comprehensive review on estimation strategies used in hybrid and battery electric vehicles," Renewable and Sustainable Energy Reviews 42 (2015) 517–531.
- [34] S. Piller, M. Perrin, and A. Jossen, "Methods for state-of-charge determination and their applications," J. Power Sources, vol. 96, no. 1, pp. 113–120, Jun. 2001.
- [35] P. J. Blood and S. Sotiriopoulos, "An electrochemical technique for state of charge (SOC) probing of positive lead-acid battery plates," J. Power Sources, vol. 110, no. 1, pp. 96–106, Jul. 2002.
- [36] V. Pop, H. J. Bergveld, P. H. L. Notten, and P. P. L. Regtien, "State-of-the-art of battery state-of-charge determination," Meas. Sci. Technol., vol. 16, no. 12, pp. R93–R110, Dec. 2005.
- [37] G. C. Hsieh, L. R. Chen, and K. S. Huang, "Fuzzy-controlled Li-ion battery charge system with active state-of-charge controller," IEEE Trans. Ind. Electron., vol. 48, no. 3, pp. 585–593, Jun. 2001.
- [38] A. H. Anbuky and P. E. Pascoe, "VRLA battery state-of-charge estimation in telecommunication power systems," IEEE Trans. Ind. Electron., vol. 47, no. 3, pp. 565–573, Jun. 2000.

- [39] O. Gerard, J. N. Patillon, and F. d'Alche-Buc, "Neural network adaptive modeling of battery discharge behavior," in *Lecture Notes in Computer Science*. Berlin, Germany: Springer-Verlag, 1997, pp. 1095–1100.
- [40] F. Huet, "A review of impedance measurements for determination of state-of-charge or state-of-health of secondary batteries," *J. Power Sources*, vol. 70, no. 1, pp. 59–69, Jan. 1998.
- [41] S. Rodrigues, N. Munichandraiah, and A. K. Shukla, "A review of state-of-charge indication of batteries by means of ac impedance measurements," *J. Power Sources*, vol. 87, no. 1/2, pp. 12–20, Apr. 2000.
- [42] L. R. Chen, "A design of an optimal battery pulse charge system by frequency-varied technique," *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 398–405, Feb. 2007.
- [43] M. Coleman, C. K. Lee, C. Zhu, and W. G. Hurley, "State-of-charge determination from EMF voltage estimation: Using impedance, terminal voltage, and current for lead-acid and lithium-ion batteries," *IEEE Trans. Ind. Electron.*, vol. 54, no. 5, pp. 2550–2557, Oct. 2007.
- [44] A. J. Salkind, C. Fennie, P. Singh, T. Atwater, and D. E. Reisner, "Determination of state-of-charge and state-of-health of batteries by fuzzy logic methodology," *J. Power Sources*, vol. 80, no. 1/2, pp. 293–300, Jul. 1999.
- [45] P. Singh, C. Fennie, and D. E. Reisner, "Fuzzy logic modeling of state-of-charge and available capacity of nickel/metal hydride batteries," *J. Power Sources*, vol. 136, no. 2, pp. 322–333, Oct. 2004.
- [46] S. Pang, J. Farrell, J. Du, and M. Barth, "Battery state-of-charge estimation," in *Proc. ACC*, Arlington, MA, 2001, pp. 1644–1649.
- [47] A. Vasebi, S. M. T. Bathae, and M. Partovibakhsh, "Predicting state of charge of lead-acid batteries for hybrid electric vehicles by extended Kalman filter," *Energy Convers. Manage.*, vol. 49, no. 1, pp. 75–82, Jan. 2008.
- [48] O. Barbarisi, F. Vasca, and L. Glielmo, "State of charge Kalman filter estimator for automotive batteries," *Control Eng. Pract.*, vol. 14, no. 3, pp. 267–275, Mar. 2006.
- [49] S. Santhanagopalan and R. E. White, "Online estimation of the state of charge of a lithium ion cell," *J. Power Sources*, vol. 161, no. 2, pp. 1346–1355, Oct. 2006.
- [50] E. Meissner and G. Richter, "Vehicle electric power systems are under change: Implications for design, monitoring and management of automotive batteries," *J. Power Sources*, vol. 95, no. 12, pp. 13–23, Mar. 2001.
- [51] G. L. Plett, "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs, Part 2: Modeling and identification," *J. Power Sources*, vol. 134, no. 2, pp. 262–276, Aug. 2004.
- [52] G. L. Plett, "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs, Part 3: State and parameter estimation," *J. Power Sources*, vol. 134, no. 2, pp. 277–292, Aug. 2004.
- [53] G. L. Plett, "Kalman-filter SOC estimation for LiPB HEV cells," in *Proc. 19th Int. Battery, Hybrid Fuel Cell Electr. Veh. Symp. Exhib.*, Bousan, Korea, pp. 1–12, 2002.
- [54] M. Charkhgard, and M. Farrokhi, "State-of-Charge Estimation for Lithium-Ion Batteries Using Neural Networks and EKF," *IEEE Trans. Ind. Electron.*, vol. 57, no. 12, pp. 4178–4187, Dec. 2010.
- [55] T. Yamazaki, K. Sakurai, and K. I. Muramoto, "Estimation of the residual capacity of sealed lead-acid batteries by neural network," in *Proc. IEEE ITEC*, San Francisco, CA, pp. 210–214, 1998.
- [56] J. Peng, Y. Chen, and R. Eberhart, "Battery pack state of charge estimator design using computational intelligence approaches," in *Proc. IEEE ABAC*, Long Beach, CA, pp. 173–177, 2000.
- [57] S. Grewal and D. A. Grant, "A novel technique for modeling the state of charge of lithium ion batteries using artificial neural networks," in *Proc. IEEE ITEC*, pp. 174–179, 2001.

- [58] C. H. Cal, D. Du, Z. Y. Liu, and H. Zhang, "Artificial neural network in estimation of battery state-of-charge (SOC) with non-conventional input variables selected by correlation analysis," in Proc. IEEE MLC, Beijing, China, pp. 1619–1625, 2002.
- [59] C. Cai, D. Du, Z. Liu, and J. Ge, "State-of-charge (SOC) estimation of high power NI-MH rechargeable battery with artificial neural network," in Proc. IEEE NIP, Singapore, pp. 824–828, 2002.
- [60] W. X. Shen, C. C. Chan, E. W. C. Lo, and K. T. Chau, "Adaptive neuro-fuzzy modeling of battery residual capacity for electric vehicles," IEEE Trans. Ind. Electron., vol. 49, no. 3, pp. 677–684, Jun. 2002.
- [61] A. Affanni, A. Bellini, C. Concari, G. Franceschini, E. Lorenzani, and C. Tassoni, "EV battery state of charge: Neural network based estimation," in Proc. IEEE ICEDM, vol. 2, pp. 684–688, 2003.
- [62] Y.-S. Lee, W.-Y. Wang, and T.-Y. Kuo, "Soft computing for battery state-of-charge (BSOC) estimation in battery string systems," IEEE Trans. Ind. Electron., vol. 55, no. 1, pp. 229–239, Jan. 2008.
- [63] C. H. Cai, D. Du, and Z. Y. Liu, "Battery state-of-charge (SOC) estimation using adaptive neuro-fuzzy inference system (ANFIS)," in Proc. IEEE ICFS, St. Louis, MO, pp. 1068–1073, 2003.
- [64] F. Codeca, S. M. Savaresi, and G. Rizzoni, "On battery State of Charge estimation: a new mixed algorithm," IEEE Int. Conf. Control Applications, pp. 102–107, Sep. 2008.
- [65] H. Blanke, O. Bohlen, S. Buller, et al., "Impedance measurements on lead-acid batteries for state-of-charge, state-of-health and cranking capability prognosis in electric and hybrid electric vehicles," J. Power Sources, no. 144, pp. 418–425, 2005.
- [66] D. Berndt, Maintenance-Free Batteries. Baldock, Hertfordshire, England: Research Studies Press LTD., 2003.
- [67] B. Hariprakash, S.K. Martha, A. Jaikumar, and A.K. Shukla, "On-line monitoring of lead-acid batteries by galvanostatic non-destructive technique," J. Power Sources, vol. 137, pp. 128–133, 2004.
- [68] D. Haifeng, W. Xuezhe, and Sun Zechang, "A New SOH Prediction Concept for the Power Lithium-ion Battery Used on HEVs", Vehicle Power and Propulsion Conf. (VPPC), 2009.
- [69] C. R. Gould, C. M. Bingham, D. A. Stone, and P. Bentley, "Battery Health Determination by Subspace Parameter Estimation and Sliding Mode Control for an All-Electric Personal Rapid Transit Vehicle – The ULTra," Power Electron. Special. Conf. (PESC), 2008.
- [70] L. Ljung, System identification – Theory for the user, 2nd ed., PTR Prentice Hall, pp. 317–353, 1999.
- [71] B. S. Bhangu, P. Bentley, D. A. Stone, C. M. Bingham, "Observer techniques for estimating the State-of-Charge and State-of-Health of VRLABs for Hybrid Electric Vehicles," IEEE Conf. Vehicle Power and Propulsion, 2005.
- [72] F. Zhang, G. Liu, and L. Fang, "A Battery State of Charge Estimation Method using Sliding Mode Observer," in Proc. 7th World Congress on Intelligent Control and Automation, pp. 989–994, Jun. 2008.
- [73] I.-S. Kim, "The novel state of charge estimation method for lithium battery using sliding mode observer," Journal of Power Sources, vol. 163, pp. 584–590, 2006.
- [74] C. Edwards, S.K. Spurgeon, C.P. Tan, and N. Patel, Sliding-Mode Observers, Springer-Verlag, Berlin, 2007.
- [75] A. J. Krener, and M. Q. Xiao, "Observers for linearly unobservable nonlinear systems," Systems & Control Letters, vol. 46, pp. 281–288, 2002.
- [76] C. Edwards, S.K. Spurgeon, and R.J. Patton, "Sliding-mode observers for fault detection," Automatica, vol. 36, pp. 541–553, 2000.

- [77] C. Edwards and S.K. Spurgeon, "Sliding Mode Control: Theory and Applications," Taylor & Francis, 1998.
- [78] V.I. Utkin, Sliding Modes in Control Optimization, Springer-Verlag, Berlin, 1992.
- [79] P. Spagnol, S. Rossi, and S. M. Savaresi, "Kalman Filter SOC Estimation for Li-Ion Batteries," IEEE Int. Conf. Control Applications (CCA), pp. 587–592, Sep. 2011.
- [80] H. Zhang, and M.-Y. Chow, "Comprehensive dynamic battery modeling for phev applications," in Power and Energy Society General Meeting, 2010 IEEE. IEEE, 2010, pp. 1–6.
- [81] Y.-C. Hsieh, T.-D. Lin, and R.-J. Chen, "Li-ion battery model exploring by intermittent discharging," in Renewable Energy Research and Applications (ICRERA), 2012 International Conference on. IEEE, 2012, pp. 1–5.
- [82] M. Verbrugge and B. Koch, "Generalized recursive algorithm for adaptive multiparameter regression application to lead acid, nickel metal hydride, and lithium-ion batteries," Journal of The Electrochemical Society, vol. 153, no. 1, pp. A187–A201, 2006.
- [83] Y. Hu, S. Yurkovich, Y. Guezennec, and B. Yurkovich, "A technique for dynamic battery model identification in automotive applications using linear parameter varying structures," Control Engineering Practice, vol. 17, no. 10, pp. 1190–1201, 2009.
- [84] M. Gholizadeh, F.R. Salmasi, "Estimation of state of charge, unknown nonlinearities, and state of health of a lithium-ion battery based on a comprehensive unobservable model," IEEE Transaction on Industrial Electronics, Vol. 61, No. 3, march 2014.
- [85] T. Kim, W. Qiao, and L. Qu, "Real-time state of charge and electrical impedance estimation for lithium-ion batteries based on a hybrid battery model," in Applied Power Electronics Conference and Exposition (APEC), 2013 Twenty-Eighth Annual IEEE. IEEE, 2013, pp. 563–568.
- [86] Y. Hu and S. Yurkovich, "Linear parameter varying battery model identification using subspace methods," Journal of Power Sources, vol. 196, no. 5, pp. 2913–2923, 2011.
- [87] Rui-hao L, Yu-kun S, Xiao-fu J. Battery state of charge estimation for electric vehicle based on neural network. In: IEEE Third international conference on communication software and networks (ICCSN); 2011. 493-6.
- [88] Yatsui M W, Bai H. Kalman filter based state-of-charge estimation for lithium-ion batteries in hybrid electric vehicles using pulse charging. IEEE Veh Power Propul Conf (VPPC) 2011:1–5.
- [89] Mastali M, Vazquez-Arenas J, Fraser R, Fowler M, Afshar S, Stevens M. Battery state of the charge estimation using Kalman filtering. J Power Sources 2013; 239:294–307.
- [90] T. Wu, L. Liu, Q. Xiao, Q. Cao, X. Wang, "Research on SOC Estimation Based on Second-order RC Model," TELKOMNIKA Indones. J. Electr. Eng. 10 (2012) 1667-1672.
- [91] He H, Xiong R, Guo H. Online estimation of model parameters and state-of-charge of LiFePO4 batteries in electric vehicles. Applied Energy 2012; 89:413–20.
- [92] S. Sepasi, R. Ghorbani, B. Y. Liaw, "A novel on-board state-of-charge estimation method for aged Li-ion batteries based on model adaptive EKF," Journal of Power Sources 245 (2014) 337-344.
- [93] R. Xiong, F. Sun, Z. Chen, H. He, "A data-driven multi-scale extended Kalman filtering based parameter and state estimation approach of lithium-ion polymer battery in electric vehicles," Applied Energy 113 (2014) 463–476.
- [94] M. Gholizadeh, S. R. Salmasi, "Estimation of State of Charge, Unknown Nonlinearities, and State of Health of a Lithium-Ion Battery Based on a Comprehensive Unobservable Model," IEEE Trans Ind Electron, Vol. 61, No. 3, pp. 1335-1344, 2014.
- [95] X. Chen, W. Shen, Z. Cao, A. Kapoor, A novel approach for state of charge estimation based on adaptive switching gain sliding mode observer in electric vehicles, J. Power Sources 246 (2014) 667-678.
- [96] A. Belhani, N. K. M'Sirdi, and A. Naamane, "Adaptive Sliding Mode Observer for Estimation of State of Charge," Energy Procedia 42 (2013) 377 – 386.

- [97] Y. C. Hsieh, S. P. Chou, and C. S. Moo, "Balance discharging for series-connected batteries," *IEEE Power Electron. Spec. Conf.*, pp. 2697–2702, 2004.
- [98] M. Y. Kim, J. W. Kim, C. H. Kim, S. Y. Cho, and G. W. Moon, "Automatic charge equalization circuit based on regulated voltage source for series connected lithium-ion batteries," *8th Int. Conf. Power Electron. (ECCE)*, pp. 2248–2255, May. 2011.
- [99] M. Einhorn, W. Roessler, and J. Fleig, "Improved Performance of Serially Connected Li-Ion Batteries With Active Cell Balancing in Electric Vehicles," *IEEE Trans. Veh. Technol.*, vol. 60, no. 6, pp. 2448–2457, Jul. 2011.
- [100] W. F. Bentley, "Cell balancing considerations for lithium-ion battery systems," in *Battery Conference on Applications and Advances*, pp. 223–226, 1997.
- [101] C. Speltino, A. Stefanopoulou, and G. Fiengo, "Cell equalization in battery stacks through State Of Charge estimation polling," in *American Control Conference (ACC)*, pp. 5050–5055, 2010.
- [102] Y. C. Hsieh, S. P. Chou, and C. S. Moo, "Balance discharging for series-connected batteries," in *Power Electronics Specialists Conference*, vol. 4, pp. 2697–2702, 2004.
- [103] C. H. Kim, M. Y. Kim, and G. W. Moon, "A Modularized Charge Equalizer Using a Battery Monitoring IC for Series-Connected Li-Ion Battery Strings in Electric Vehicles," *IEEE Trans. Power Electron.*, vol. 28, no. 8, pp. 3779–3787, Aug. 2013.
- [104] B. T. Kuhn, G. E. Pitel, and P. T. Krein, "Electrical properties and equalization of lithium-ion cells in automotive applications," in *Vehicle Power and Propulsion IEEE Conference*, 2005.
- [105] X. Wei and B. Zhu, "The research of vehicle power Li-ion battery pack balancing method," pp. 2498–2502, 2009.
- [106] J. Zhao, J. Jiang, and L. Niu, "A novel charge equalization technique for electric vehicle battery system," *International Conference on Power Electronics and Drive Systems*, vol. 2, pp. 853–857, 2003.
- [107] N. H. Kutkut, H. L. N. Wiegman, D. M. Divan, and D. W. Novotny, "Charge equalization for an electric vehicle battery system," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 34, no. 1, pp. 235–246, Jan. 1998.
- [108] J. W. Kimball, B. T. Kuhn, and P. T. Krein, "Increased Performance of Battery Packs by Active Equalization," in *IEEE Vehicle Power and Propulsion Conference*, pp. 323–327, 2007.
- [109] A. Baughman and M. Ferdowsi, "Double-Tiered Switched-Capacitor Battery Charge Equalization Technique," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2277–2285, Jun. 2008.
- [110] T. Stuart, F. Fang, X. Wang, C. Ashtiani, and A. Pesaran, "A Modular Battery Management System for HEVs," *SAE*, pp. 1918-1923, Jun. 2002.
- [111] Y. S. Lee and G. T. Cheng, "Quasi-Resonant Zero-Current-Switching Bidirectional Converter for Battery Equalization Applications," *IEEE Trans. Power Electron.*, vol. 21, no. 5, pp. 1213–1224, Sep. 2006.
- [112] Y. S. Lee and M. W. Cheng, "Intelligent control battery equalization for series connected lithium-ion battery strings," *IEEE Trans. Ind. Electron.*, vol. 52, no. 5, pp. 1297–1307, Oct. 2005.
- [113] K. Zhi-guo, Z. Chun-bo, L. Ren-gui, and C. Shu-kang, "Comparison and Evaluation of Charge Equalization Technique for Series Connected Batteries," in *37th IEEE Power Electronics Specialists Conference*, pp. 1–6, 2006.
- [114] C. Bonfiglio and W. Roessler, "A cost optimized battery management system with active cell balancing for lithium ion battery stacks," in *IEEE Vehicle Power and Propulsion Conference. VPPC*, pp. 304–309, 2009.
- [115] J. W. Shin, G. S. Seo, C.Y. Chun, and B. H. Cho, "Selective flyback balancing circuit with improved balancing speed for series connected Lithium-ion batteries," in *Power Electronics Conference (IPEC)*, pp. 1180–1184, 2010.

- [116] A. Imtiaz and F. H. Khan, "Time Shared Flyback Converter Based Regenerative Cell Balancing Technique for Series Connected Li-Ion Battery Strings," *IEEE Trans. Power Electron.*, vol. 28, no. 12, pp. 5960–5975, Dec. 2013.

۲- طراحی و شبیه‌سازی روش‌های مورد بررسی

۲-۱- مقدمه

در این پروژه قرار است دو روش یکسان‌سازی شارژ فعال و غیرفعال برای یکسان‌سازی SOC در چهار سلول سری مورد استفاده قرار گیرد. از آنجایی که SOC به طور مستقیم قابل اندازه‌گیری نیست لذا باید تخمین زده شود. روش‌های مختلفی برای تخمین SOC می‌توان به کار گرفت. در این پروژه از دو روش رؤیتگر مد لغزشی (SMO)^۱ و فیلتر کالمن توسعه یافته (EKF)^۲ به علت خواصی چون مقاوم بودن نسبت به نویز و تغییرات پارامتریک، استفاده می‌شود. در این فصل به تشریح کدهای مربوط به شبیه‌سازی روش‌های یکسان‌سازی شارژ فعال و غیرفعال بر مبنای هر دو روش تخمین SMO و EKF در محیط نرم‌افزار MATLAB پرداخته است. نحوه سازماندهی مطالب فصل در ادامه به این صورت است که ابتدا به معرفی روش‌های استفاده شده برای تخمین SOC باتری، یعنی SMO و EKF و نحوه شبیه‌سازی این روش‌ها در MATLAB اختصاص یافته است. استخراج معادلات فضای حالت مدل الکتریکی باتری نیز در ادامه انجام می‌گیرد. سپس نحوه شبیه‌سازی روش یکسان‌سازی غیرفعال سطح شارژ در نرم‌افزار MATLAB تشریح گردیده و از آنجایی که برای شبیه‌سازی این روش، یک بار از روش SMO برای تخمین حالت‌های باتری از جمله SOC استفاده شده است و بار دیگر از روش EKF، بنابراین دو بخش مجزا برای توضیح هر کدام اختصاص داده شده است. در ادامه کد مربوط به شبیه‌سازی روش یکسان‌سازی فعال سطح شارژ در MATLAB تشریح می‌گردد.

۲-۲- شبیه‌سازی الگوریتم‌های تخمین SOC در MATLAB

همانطور که قبلاً گفته شد، استفاده از اطلاعات سطح شارژ (SOC) سلول‌ها در الگوریتم‌های یکسان‌سازی سطح شارژ سلول‌های یک باتری، بر دقت و کارایی این الگوریتم‌ها می‌افزاید. از آنجا که پارامتر SOC بصورت مستقیم قابل اندازه‌گیری نیست، باید تخمینی از آن در الگوریتم‌های یکسان‌سازی استفاده شود. در گزارش فاز اول پروژه [۷]، دو تخمین‌گر فیلتر کالمن تعمیم‌یافته (EKF) و رؤیتگر مد لغزشی (SMO) به منظور تخمین SOC معرفی و روابط حاکم بر آنها تشریح گردید. در این فصل با پیاده‌سازی الگوریتم‌های هر یک از این دو

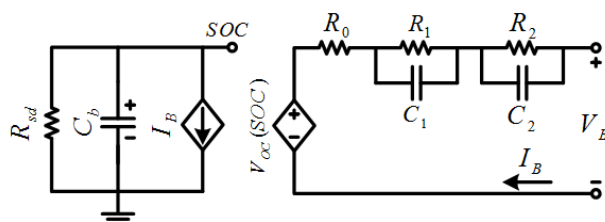
¹ Sliding Mode Observer

² Extended Kalman Filter

روش در محیط MATLAB به شبیه‌سازی عملکرد و ارزیابی صحت و دقت این روش‌ها پرداخته می‌شود. به این منظور ابتدا باید یک مدل دینامیکی از رفتار باتری تحت جریان‌های بار و مقادیر مختلف پارامترهای مشخصه باتری در اختیار داشت. در ادامه ابتدا به تشریح روش مدل‌سازی مورد نظر پرداخته سپس کد پیاده‌سازی هریک از الگوریتم‌های تخمین در محیط MATLAB ارائه می‌شود. لازم به ذکر است که در اینجا فرض شده است که باتری از ۴ سلول تشکیل شده است.

۱-۲-۲- مدل‌سازی باتری

مدل انتخابی باتری، مدل رندل مرتبه دوم می‌باشد که پیش‌تر در فصل دوم از گزارش اول معرفی شد [۷]. این مدل در شکل ۱-۲ آمده است.



شکل ۱-۲- مدل الکتریکی باتری

مقاومت R_{sd} مقاومت اهمی باتری و شبکه‌های RC برای مدل کردن رفتار پاسخ گذاری ولتاژ باتری است. R_{sd} بیانگر مقاومت دشارژ خودبه‌خودی و $C_b = 3600C_n$ خازن ظرفیت کامل است که C_n همان ظرفیت آمپرساعت باتری است. منبع ولتاژ کنترل شده در این مدل، ولتاژ مدار باز باتری، V_{OC} ، است که خود تابعی غیرخطی از SOC است. V_{OC} به صورت زیر مدل می‌شود [۱]:

$$V_{OC}(SOC) = \alpha_0 + \alpha_1 SOC + \alpha_2 SOC^2 + \alpha_3 SOC^3 + \alpha_4 e^{-\beta SOC} \quad (2-1)$$

که در آن $SOC \in [0, 1]$ و $\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \beta$ پارامترهای ثابت هستند.

V_B و I_B به ترتیب ولتاژ ترمینال و جریان ترمینال باتری را نشان می‌دهند. ولتاژ پایانه باتری به صورت زیر بیان می‌شود:

$$V_B = V_{OC}(SOC) - R_0 I_B - V_1 - V_2 \quad (2-2)$$

از سمت چپ مدل، SOC به صورت زیر بیان می‌شود:

$$SOC = -\frac{1}{R_{sd}C_b} SOC - \frac{1}{C_b} I_B \quad (2-3)$$

روابط مربوط به ولتاژهای دو سر شبکه‌های RC به صورت زیر به دست می‌آیند:

$$\begin{cases} \dot{V}_1 = -\frac{1}{R_1 C_1} V_1 + \frac{1}{C_1} I_B \\ \dot{V}_2 = -\frac{1}{R_2 C_2} V_2 + \frac{1}{C_2} I_B \end{cases} \quad (2-4)$$

بنابراین معادلات فضای حالت باتری به صورت زیر خواهد بود:

$$\begin{cases} \dot{SOC} = -\frac{1}{R_{sd}C_b}SOC - \frac{1}{C_b}I_B \\ \dot{V}_1 = -\frac{1}{R_1C_1}V_1 + \frac{1}{C_1}I_B \\ \dot{V}_2 = -\frac{1}{R_2C_2}V_2 + \frac{1}{C_2}I_B \end{cases} \quad (2-5)$$

$$V_B = V_{OC}(SOC) - V_1 - V_2 - R_0 I_B$$

با تعریف $x = [SOC \ V_1 \ V_2]^T$ ، $u = I_B$ ، $y = V_B$ معادلات حالت و خروجی سیستم به صورت زیر نوشته می شود:

$$\begin{cases} \dot{x} = Ax + Bu \\ y = g(x, u) \end{cases} \quad (2-6)$$

که در آن داریم:

$$A = \begin{bmatrix} -\frac{1}{R_{sd}C_b} & 0 & 0 \\ 0 & -\frac{1}{R_1C_1} & 0 \\ 0 & 0 & -\frac{1}{R_2C_2} \end{bmatrix}$$

$$B = \begin{bmatrix} -\frac{1}{C_b} & \frac{1}{C_1} & \frac{1}{C_2} \end{bmatrix}^T \quad (2-7)$$

$$g(x, u) = V_{OC}(x_1) - x_2 - x_3 - R_0 u$$

فرم گسسته این معادلات در الگوریتم EKF استفاده می شود. گسسته شده این مدل با زمان نمونه برداری dt به صورت زیر خواهد بود:

$$\begin{cases} x_{k+1} = A_d x_k + B_d u_k \\ y_k = g(x_k, u_k) \end{cases} \quad (2-8)$$

که در آن $B_d = dt.B$ ، $A_d = I - dt.A$.

۲-۲-۲- شبیه سازی الگوریتم EKF در محیط MATLAB جهت تخمین سطح شارژ

سیستم غیرخطی باتری به فرم زیر را در نظر بگیرید:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{w}_k \\ \mathbf{y}_k &= \mathbf{g}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{v}_k \end{aligned} \quad (2-9)$$

که در آن $\mathbf{A}$ ، $\mathbf{B}$ ماتریس های انتقال حالت، $\mathbf{g}(\cdot, \cdot)$ تابع اندازه گیری، $\mathbf{w}_k$ نویز فرایند و $\mathbf{v}_k$ نویز اندازه گیری هستند. این نویزها، نویزهای گوسی سفید مستقل با میانگین صفر و با ماتریس های کواریانس زیر می باشند:

$$\mathbf{Q}_k = E[\mathbf{w}_k \mathbf{w}_k^T] \quad , \quad \mathbf{R}_k = E[\mathbf{v}_k \mathbf{v}_k^T] \quad (2-10)$$

برای اعمال EKF، مدل باتری باید در هر زمان نمونه برداری با استفاده از بسط سری تیلور حول آخرین تخمین $\hat{\mathbf{x}}_k$ خطی شود. مشتق گیری از تابع $\mathbf{g}(\cdot, \cdot)$ در رابطه (۲-۱۱) نسبت به $\mathbf{x}_k$ نتیجه می دهد:

$$\mathbf{C}_k = \left. \frac{\delta \mathbf{g}}{\delta \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_k} = \left[\left. \frac{dV_{OC}}{dSOC} \right|_{SOC=\hat{SOC}_k}, -1, -1 \right] \quad (2-12)$$

می توان نشان داد که ماتریس A در رابطه (۲-۷) به همراه ماتریس فوق یک جفت رؤیت پذیر می باشند. در روابط فوق $\hat{\mathbf{x}}_k$ به عنوان پس تخمین حالت $\mathbf{x}_k$ بعد از زمان k تعریف شده است که با در دست داشتن اندازه $\mathbf{y}_k$ قابل محاسبه است. همچنین $\hat{\mathbf{x}}_k^-$ به عنوان پیش تخمین حالت در زمان k تعریف شده است که با در دست داشتن اطلاعات از فرایند قبل از زمان k قابل محاسبه است. حال با در نظر گرفتن مقادیر اولیه زیر برای متغیرهای حالت و کواریانس خطا، الگوریتم EKF به صورت جدول ۲-۱ خلاصه می شود [۱۰].

$$\hat{\mathbf{x}}_0 = E[\mathbf{x}_0] \quad , \quad \mathbf{P}_0 = E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T] \quad (2-13)$$

EKF جدول ۲-۱- خلاصه الگوریتم

محاسبه به ازای $k = 1, 2, \dots$	
$\hat{\mathbf{x}}_k^- = \mathbf{f}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1})$ $\mathbf{P}_k^- = \mathbf{A}_{k-1} \mathbf{P}_{k-1} \mathbf{A}_{k-1}^T + \mathbf{Q}_{k-1}$	<u>به روز رسانی زمان:</u> انتشار تخمین حالت : انتشار کواریانس خطا :
$\mathbf{L}_k = \mathbf{P}_k^- \mathbf{C}_k^T (\mathbf{C}_k \mathbf{P}_k^- \mathbf{C}_k^T + \mathbf{R}_k)^{-1}$ $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{L}_k (\mathbf{y}_k - \mathbf{g}(\hat{\mathbf{x}}_k^-, \mathbf{u}_k))$ $\mathbf{P}_k = (\mathbf{I} - \mathbf{L}_k \mathbf{C}_k) \mathbf{P}_k^-$	<u>به روز رسانی اندازه گیری:</u> ماتریس بهره کالمن : به روز رسانی تخمین حالت: به روز رسانی کواریانس خطا:

پیاده سازی الگوریتم EKF

در این بخش، تابع States_Ich_Update_SOCbalancing_EKF که در آن از الگوریتم EKF برای تخمین SOC استفاده شده است تشریح می شود.

ابتدا ماتریس های فضای حالت مدل باتری و کوواریانس نویز فرایند Q و نویز اندازه گیری R تعریف می شوند.

$$\begin{aligned} \mathbf{A} &= [1 - dt / (R_{sd} * C_b), 0, 0; 0, 1 - dt / (R_1 * C_1), 0; 0, 0, 1 - dt / (R_2 * C_2)] ; \\ \mathbf{B} &= [dt / C_b ; dt / C_1 ; dt / C_2] ; \\ \mathbf{Q} &= 10^{-8} * \mathbf{eye}(3) ; \\ \mathbf{R} &= 10^{-6} ; \end{aligned}$$

با داشتن جریان بالانس و جریان شارژ، جریان هر سلول محاسبه می شود و سپس ولتاژ هر سلول به دست می آید.

```
I1=Ich+Ibal1;
I2=Ich+Ibal2;
I3=Ich+Ibal3;
I4=Ich+Ibal4;
V1=Voc1+V11+V12+R0*I1;
V2=Voc2+V21+V22+R0*I2;
V3=Voc3+V31+V32+R0*I3;
V4=Voc4+V41+V42+R0*I4;
```

در ادامه، معادلات فضای حالت گسسته (رابطه ۲-۸) برای هر سلول پیاده‌سازی شده و مقادیر جدید متغیرهای حالت و ولتاژ مدار باز هر سلول به دست می‌آید. این مقادیر برای محاسبه ولتاژ سلول در زمان نمونه‌برداری بعدی استفاده می‌شوند.

```
%%**** Battery states update ****
%****first states update
SOC1_new = (1-dt/(Rsd*Cb))*SOC1+I1*dt/Cb;
V11_new = (1-dt/(R1*C1))*V11+I1*dt/C1;
V12_new = (1-dt/(R2*C2))*V12+I1*dt/C2;
Voc1=a0+a1*SOC1_new+a2*SOC1_new^2+a3*SOC1_new^3+a4*exp(-b*SOC1_new);
%****second battery states update
SOC2_new = (1-dt/(Rsd*Cb))*SOC2+I2*dt/Cb;
V21_new = (1-dt/(R1*C1))*V21+I2*dt/C1;
V22_new = (1-dt/(R2*C2))*V22+I2*dt/C2;
Voc2=a0+a1*SOC2_new+a2*SOC2_new^2+a3*SOC2_new^3+a4*exp(-b*SOC2_new);
%****third battery states update
SOC3_new = (1-dt/(Rsd*Cb))*SOC3+I3*dt/Cb;
V31_new = (1-dt/(R1*C1))*V31+I3*dt/C1;
V32_new = (1-dt/(R2*C2))*V32+I3*dt/C2;
Voc3=a0+a1*SOC3_new+a2*SOC3_new^2+a3*SOC3_new^3+a4*exp(-b*SOC3_new);
%****forth battery states update
SOC4_new = (1-dt/(Rsd*Cb))*SOC4+I4*dt/Cb;
V41_new = (1-dt/(R1*C1))*V41+I4*dt/C1;
V42_new = (1-dt/(R2*C2))*V42+I4*dt/C2;
Voc4=a0+a1*SOC4_new+a2*SOC4_new^2+a3*SOC4_new^3+a4*exp(-b*SOC4_new);
```

در گام بعد، معادلات EKF جهت بهبود تخمین حالت‌ها برای چهار سلول اعمال می‌گردد. در اینجا، متغیرهای تخمین زده شده با حرف h همراه نام متغیر مشخص شده‌اند. متغیرهای SOChi و Vhi1 و Vhi2 {i=1,2,3,4} به ترتیب تخمین SOC، تخمین ولتاژ شبکه RC اول و تخمین ولتاژ شبکه RC دوم در مدل الکتریکی سلول نام می‌باشند. بردار Xhi تخمین متغیرهای حالت سلول نام می‌باشد. بردار H از رابطه (۲-۱۱) محاسبه می‌شود و ماتریس K همان بهره کالمن در جدول ۱-۲ می‌باشد. متغیر OCV، ولتاژ مدار باز و Vhi تخمین ولتاژ ترمینال سلول نام است. معادلات Xhi و Pi در واقع همان بروزرسانی تخمین حالت و کوواریانس

خطا می‌باشند که از ادغام معادلات مربوط به بروزرسانی زمان و بروزرسانی اندازه‌گیری در جدول ۱-۲ حاصل می‌شوند.

```

%%***** EKF Equations *****
%****first battery states update
Xh1 = [SOCh1;Vh11;Vh12];
H = [a1+2*a2*SOCh1+3*a3*SOCh1^2-b*a4*exp(-b*SOCh1) , 1 , 1];
K = P1*H' / (H*P1*H'+R);
OCV = a0+a1*SOCh1+a2*SOCh1.^2+a3*SOCh1.^3+a4*exp(-b*SOCh1);
Vh1 = OCV+Vh11+Vh12+R0*I1;
Xh1 = A*Xh1 + B*I1 + A*K*(V1-Vh1);
P1 = A*(eye(3)-K*H)*P1*A' + Q;
SOCh1_new = Xh1(1); Vh11_new = Xh1(2); Vh12_new = Xh1(3);
%****second battery states update
Xh2 = [SOCh2;Vh21;Vh22];
H = [a1+2*a2*SOCh2+3*a3*SOCh2^2-b*a4*exp(-b*SOCh2) , 1 , 1];
K = P2*H' / (H*P2*H'+R);
OCV = a0+a1*SOCh2+a2*SOCh2.^2+a3*SOCh2.^3+a4*exp(-b*SOCh2);
Vh2 = OCV+Vh21+Vh22+R0*I2;
Xh2 = A*Xh2 + B*I2 + A*K*(V2-Vh2);
P2 = A*(eye(3)-K*H)*P2*A' + Q;
SOCh2_new = Xh2(1); Vh21_new = Xh2(2); Vh22_new = Xh2(3);
%****third battery states update
Xh3 = [SOCh3;Vh31;Vh32];
H = [a1+2*a2*SOCh3+3*a3*SOCh3^2-b*a4*exp(-b*SOCh3) , 1 , 1];
K = P3*H' / (H*P3*H'+R);
OCV = a0+a1*SOCh3+a2*SOCh3.^2+a3*SOCh3.^3+a4*exp(-b*SOCh3);
Vh3 = OCV+Vh31+Vh32+R0*I3;
Xh3 = A*Xh3 + B*I3 + A*K*(V3-Vh3);
P3 = A*(eye(3)-K*H)*P3*A' + Q;
SOCh3_new = Xh3(1); Vh31_new = Xh3(2); Vh32_new = Xh3(3);
%****forth battery states update
Xh4 = [SOCh4;Vh41;Vh42];
H = [a1+2*a2*SOCh4+3*a3*SOCh4^2-b*a4*exp(-b*SOCh4) , 1 , 1];
K = P4*H' / (H*P4*H'+R);
OCV = a0+a1*SOCh4+a2*SOCh4.^2+a3*SOCh4.^3+a4*exp(-b*SOCh4);
Vh4 = OCV+Vh41+Vh42+R0*I4;
Xh4 = A*Xh4 + B*I4 + A*K*(V4-Vh4);
P4 = A*(eye(3)-K*H)*P4*A' + Q;
SOCh4_new = Xh4(1); Vh41_new = Xh4(2); Vh42_new = Xh4(3);

```

در ادامه، جریان شارژ باتری تعیین می‌شود. همانطور که قبلاً اشاره شد فرایند یکسان‌سازی تا جایی ادامه می‌یابد که باتری کاملاً شارژ شود که شرط شارژ کامل، رسیدن جریان شارژ به یک حد کمینه (در اینجا 1mA) است. فرایند شارژ باتری به دو بخش جریان ثابت (CC)¹ و ولتاژ ثابت (CV)² تقسیم می‌شود. تا زمانی که ولتاژ مجموعه باتری به یک مقدار مشخص (معمولاً این مقدار برابر حداکثر ولتاژ هر سلول ضربدر تعداد سلول‌ها است که در اینجا V 16.4 خواهد بود) نرسد جریان شارژ باتری ثابت است. پس از این مرحله، ولتاژ شارژ ثابت می‌ماند و جریان بتدریج کاهش می‌یابد تا به آن حد کمینه 1mA برسد. کد زیر، این فرایند دومرحله‌ای شارژ را بیان می‌کند.

```
if Ich>0.001
    if (V1+V2+V3+V4)<16.4 && max([V1;V2;V3;V4])<4.1
        %%% CC charging %%%
        dIch=0;
    else
        %%% CV charging %%%
        dVoc1dSOC1 = (a1+2*a2*SOCh1+3*a3*SOCh1^2-b*a4*exp(-b*SOCh1));
        dVoc2dSOC2 = (a1+2*a2*SOCh2+3*a3*SOCh2^2-b*a4*exp(-b*SOCh2));
        dVoc3dSOC3 = (a1+2*a2*SOCh3+3*a3*SOCh3^2-b*a4*exp(-b*SOCh3));
        dVoc4dSOC4 = (a1+2*a2*SOCh4+3*a3*SOCh4^2-b*a4*exp(-b*SOCh4));
        Voc1dot = dVoc1dSOC1*(SOCh1_new-SOCh1)/dt;
        Voc2dot = dVoc2dSOC2*(SOCh2_new-SOCh2)/dt;
        Voc3dot = dVoc3dSOC3*(SOCh3_new-SOCh3)/dt;
        Voc4dot = dVoc4dSOC4*(SOCh4_new-SOCh4)/dt;
        V11dot = (Vh11_new-Vh11)/dt; V12dot = (Vh12_new-Vh12)/dt;
        V21dot = (Vh21_new-Vh21)/dt; V22dot = (Vh22_new-Vh22)/dt;
        V31dot = (Vh31_new-Vh31)/dt; V32dot = (Vh32_new-Vh32)/dt;
        V41dot = (Vh41_new-Vh41)/dt; V42dot = (Vh42_new-Vh42)/dt;
        dIch=-
        dt/(4*R0)*(Voc1dot+Voc2dot+Voc3dot+Voc4dot+V11dot+V12dot+
        V21dot+V22dot+V31dot+V32dot+V41dot+V42dot);
    end
    Ich_new=Ich+dIch;
else
    Ich_new=0;
end
```

در کد فوق، تغییرات جریان شارژ در مرحله CV با مشتق‌گیری از رابطه ولتاژ ترمینال هر سلول به دست می‌آید:

¹ Constant Current

² Constant Voltage

$$I_B = -\frac{1}{R_0}(V_{OC}(SOC) + V_1 + V_2 - V_B) \Rightarrow \dot{I}_B = -\frac{1}{R_0} \left(\frac{\partial V_{OC}}{\partial SOC} \dot{SOC} + \dot{V}_1 + \dot{V}_2 - \dot{V}_B \right) \quad (2-14)$$

از آنجایی که در مرحله CV منبع ولتاژ ثابت برای شارژ داریم، $\dot{V}_B = 0$ خواهد بود. همچنین در رابطه فوق، مقادیر مشتق متغیرها، با در دست داشتن مقادیر قبلی و فعلی تخمین آن متغیرها، با استفاده از رابطه اولیو به دست می‌آید. به عنوان مثال برای SOC داریم:

$$\dot{SOC} = (\hat{SOC}_k - \hat{SOC}_{k-1}) / dt \quad (2-15)$$

۲-۲-۳- شبیه‌سازی الگوریتم SMO در MATLAB

برای تخمین حالت با رؤیتگر مد لغزشی، ابتدا ولتاژ ترمینال را به عنوان یک حالت در نظر می‌گیریم. با مشتق‌گیری از خروجی V_B در رابطه (۲-۵) و با این فرض که در هر دوره نمونه‌برداری در حالت زمان گسسته داریم، $\frac{dI_B}{dt} \approx 0$

$$\begin{aligned} \dot{V}_B &= \frac{\partial V_{OC}}{\partial SOC} \dot{SOC} - \dot{V}_1 - \dot{V}_2 \\ &= (\alpha_1 + 2\alpha_2 SOC + 3\alpha_3 SOC^2 - \beta\alpha_4 e^{-\beta SOC}) \left(-\frac{1}{R_{sd}C_b} SOC - \frac{1}{C_b} I_B \right) \\ &\quad + \frac{1}{R_1 C_1} V_1 - \frac{1}{C_1} I_B + \frac{1}{R_2 C_2} V_2 - \frac{1}{C_2} I_B \\ &= \left[-\frac{\alpha_1}{R_{sd}C_b} - \frac{1}{R_{sd}C_b} (2\alpha_2 SOC + 3\alpha_3 SOC^2 - \beta\alpha_4 e^{-\beta SOC}) \right] SOC - \frac{1}{R_1 C_1} V_1 \\ &\quad + \frac{1}{R_2 C_2} V_2 + \left[-\frac{\alpha_1}{C_b} - \frac{1}{C_b} (2\alpha_2 SOC + 3\alpha_3 SOC^2 - \beta\alpha_4 e^{-\beta SOC}) - \frac{1}{C_1} - \frac{1}{C_2} \right] I_B \end{aligned} \quad (2-16)$$

به جای V_2 قرار می‌دهیم: $V_2 = -V_B + V_{OC}(SOC) - V_1 - R_0 I_B$

$$\begin{aligned} \dot{V}_B &= \left[-\frac{\alpha_1}{R_{sd}C_b} - \frac{1}{R_{sd}C_b} (2\alpha_2 SOC + 3\alpha_3 SOC^2 - \beta\alpha_4 e^{-\beta SOC}) \right] SOC + \frac{1}{R_1 C_1} V_1 \\ &\quad + \frac{1}{R_2 C_2} (-V_B + \alpha_0 + \alpha_1 SOC + \alpha_2 SOC^2 + \alpha_3 SOC^3 + \alpha_4 e^{-\beta SOC} - V_1 - R_0 I_B) \\ &\quad + \left[-\frac{\alpha_1}{C_b} - \frac{1}{C_b} (2\alpha_2 SOC + 3\alpha_3 SOC^2 - \beta\alpha_4 e^{-\beta SOC}) - \frac{1}{C_1} - \frac{1}{C_2} \right] I_B \\ &= \left(-\frac{\alpha_1}{R_{sd}C_b} + \frac{\alpha_1}{R_2 C_2} \right) SOC + \left(\frac{1}{R_1 C_1} - \frac{1}{R_2 C_2} \right) V_1 - \frac{1}{R_2 C_2} V_B \\ &\quad + \left(\frac{-R_0}{R_2 C_2} - \frac{\alpha_1}{C_b} - \frac{1}{C_1} - \frac{1}{C_2} \right) I_B + f_{NL}(SOC, I_B) \end{aligned} \quad (2-17)$$

که در آن داریم:

$$f_{NL}(SOC, I_B) = \left(2\alpha_2 SOC + 3\alpha_3 SOC^2 - \beta\alpha_4 e^{-\beta SOC} \right) \left(-\frac{1}{R_{sd}C_b} SOC - \frac{1}{C_b} I_B \right) + \frac{1}{R_2 C_2} (\alpha_0 + \alpha_2 SOC^2 + \alpha_3 SOC^3 + \alpha_4 e^{-\beta SOC}) \quad (2-18)$$

در واقع بخش خطی معادله (۱-۲) با خطی سازی V_{OC} به صورت رابطه زیر به دست می‌آید:

$$V_{OC}(SOC) = \alpha_1 SOC + d \quad (2-19)$$

که در آن d خطای خطی سازی است که از حذف ترم‌های مرتبه بالا و نمایی حاصل شده است.

در نتیجه معادلات حالت و خروجی سیستم به صورت زیر خواهند بود [۷]:

$$\begin{aligned} x &= [SOC \quad V_1 \quad V_2 \quad V_B]^T, \quad u = I_B, \quad y = V_B \\ \begin{cases} \dot{x} = Ax + Bu + Tf(x, u) \\ y = Cx \end{cases} \\ A &= \begin{bmatrix} -\frac{1}{R_{sd}C_b} & 0 & 0 & 0 \\ 0 & -\frac{1}{R_1C_1} & 0 & 0 \\ 0 & 0 & -\frac{1}{R_2C_2} & 0 \\ \frac{-\alpha_1}{R_{sd}C_b} + \frac{\alpha_1}{R_2C_2} & \frac{1}{R_1C_1} - \frac{1}{R_2C_2} & 0 & -\frac{1}{R_2C_2} \end{bmatrix} \end{aligned} \quad (2-20)$$

$$B = \begin{bmatrix} \frac{1}{C_b} & \frac{1}{C_1} & \frac{1}{C_2} & -\frac{R_0}{R_2C_2} - \frac{\alpha_1}{C_b} - \frac{1}{C_1} - \frac{1}{C_2} \end{bmatrix}^T, \quad C = [0 \quad 0 \quad 0 \quad 1]$$

از آنجایی که مدل خطی باتری در مقایسه با سلول واقعی باتری دقیق نیست، بنابراین ترم‌های اغتشاشی $Tf(x, u)$ برای جبران سازی خطاهای مدل سازی و خطی سازی به مدل اضافه می‌شود. پارامتر T ماتریس ورودی اغتشاشی و تابع $f \in \mathbb{R}^q$ محدود فرض می‌شود، که $\|f\| \leq h_0$ ، که h_0 ثابت است.

فرم گسسته این معادلات در الگوریتم SMO استفاده می‌شود. گسسته شده این مدل با زمان نمونه برداری dt به صورت زیر خواهد بود:

$$\begin{cases} x_{k+1} = A_d x_k + B_d u_k + dt.Tf(x_k, u_k) \\ y_k = C_d x_k \end{cases} \quad (2-21)$$

که در آن $C_d = C$, $B_d = dt.B$, $A_d = I + dt.A$.

رویتگر SMO برای سیستم پیوسته به صورت زیر داده می‌شود [۷]:

$$\begin{cases} \dot{\hat{x}} = A\hat{x} + Bu + H(y - \hat{y}) + hTv(e_y) \\ \hat{y} = C\hat{x} \end{cases} \quad (2-22)$$

که در آن ماتریس بهره رؤیتگر H و بهره سوئیچینگ h طوری انتخاب می‌شوند که پایداری SMO تضمین شود. ترم $hT v(e_y)$ قاعده سوئیچینگ ناپیوسته را نشان می‌دهد که در آن $v(e_y)$ سیگنال تزریق خطای خروجی $e_y = y - \hat{y}$ می‌باشد و به صورت زیر تعریف می‌شود:

$$v(e_y) = \text{sgn}(We_y) = \begin{cases} +1 & e_y > 0 \\ -1 & e_y < 0 \end{cases} \quad (2-23)$$

که در آن ماتریس W یک ماتریس مثبت معین می‌باشد که در ادامه معرفی می‌شود. ماتریس بهره رؤیتگر H با استفاده از روش جابجایی قطب تعیین می‌شود. شرط لازم و کافی وجود بهره رؤیتگر H برای جابجایی قطب‌های رؤیت‌گر، رؤیت‌پذیری سیستم است. با بررسی رؤیت‌پذیری جفت (A, C) خواهیم دید که سیستم رؤیت‌ناپذیر است. با توجه به رؤیت‌ناپذیر بودن سیستم، نمی‌توان H را طوری تعیین کرد که بتوان همه حالت‌های سیستم را تخمین زد. برای حل این مسأله، آن را به مدهای رؤیت‌پذیر و رؤیت‌ناپذیر تفکیک می‌کنیم. با این کار، مدهای رؤیت‌پذیر سیستم را می‌توان تخمین زد. با تجزیه سیستم به مدهای رؤیت‌پذیر و مدهای رؤیت‌ناپذیر، معادلات سیستم به فرم زیر درمی‌آیند:

$$\begin{cases} \begin{bmatrix} \dot{x}_o \\ \dot{x}_u \end{bmatrix} = \begin{bmatrix} A_o & 0 \\ 0 & A_u \end{bmatrix} \begin{bmatrix} x_o \\ x_u \end{bmatrix} + \begin{bmatrix} B_o \\ B_u \end{bmatrix} u + Tf(x, u) \\ y = \begin{bmatrix} C_o & 0 \end{bmatrix} \begin{bmatrix} x_o \\ x_u \end{bmatrix} \end{cases} \quad (2-24)$$

که در آن x_o مدهای رؤیت‌پذیر و x_u مدهای رؤیت‌ناپذیر سیستم بوده و ماتریس‌های A_o, A_u, C_o به صورت زیر می‌باشند:

$$A_o = \begin{bmatrix} -\frac{1}{R_{sd}C_b} & 0 & 0 \\ 0 & -\frac{1}{R_1C_1} & 0 \\ -\frac{\alpha_1}{R_{sd}C_b} + \frac{\alpha_1}{R_2C_2} & \frac{1}{R_1C_1} - \frac{1}{R_2C_2} & -\frac{1}{R_2C_2} \end{bmatrix}, A_u = \begin{bmatrix} -\frac{1}{R_2C_2} \end{bmatrix}, C_o = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \quad (2-25)$$

از آنجایی که قطب رؤیت‌ناپذیر سیستم (مقدار ویژه ماتریس A_u) پایدار می‌باشد، لذا سیستم آشکاری‌پذیر می‌باشد.

ماتریس بهره رؤیتگر به صورت زیر در نظر گرفته می‌شود:

$$H = \begin{bmatrix} H_o \\ 0 \end{bmatrix} \quad (2-26)$$

برای زوج رؤیت پذیر (A_o, C_o) می توان ماتریس بهره رؤیتگر H_o را به روش جایی قطب تعیین کرد. لذا به ازای ماتریس مثبت معین Q_o ، یک جواب منحصر به فرد مثبت معین P_o برای معادله لیاپانوف زیر وجود دارد:

$$P_o (A_o - H_o C_o) + (A_o - H_o C_o)^T P_o = -Q_o \quad (2-27)$$

ماتریس P را به صورت زیر تعریف می کنیم:

$$P = \begin{bmatrix} P_o & 0 \\ 0 & 1 \end{bmatrix} \quad (2-28)$$

حال فرض می کنیم که به ازای ماتریس مثبت معین P ، شرط زیر به ازای یک ماتریس مثبت معین W برآورده می شود:

$$WC = T^T P \quad (2-29)$$

این ماتریس W ، همان ماتریس معرفی شده در رابطه (۲۲-۲) می باشد. در اینجا معادلات رؤیتگر به فرم گسسته پیاده سازی شده است که گسسته شده رابطه (۲۱-۲) با زمان نمونه برداری dt می باشد:

$$\begin{cases} \hat{x}_{k+1} = A_d \hat{x}_k + dt.Bu_k + dt.H(y_k - \hat{y}_k) + dt.hTv(e_{y_k}) \\ \hat{y}_k = C_d \hat{x}_k \end{cases} \quad (2-30)$$

که در آن $A_d = I + dt.A$ ، $C_d = C$.

پیاده سازی الگوریتم SMO

در این بخش، تابع States_Ich_Update_SOCbalancing_SMO که در آن از الگوریتم SMO برای تخمین SOC استفاده شده است تشریح می شود.

همانطور که دیده شد، معادلات فضای حالت (۱۹-۲) با خطی سازی منحنی غیرخطی $V_{OC}(SOC)$ استخراج شده است. در واقع یک منحنی خطی به فرم زیر به این منحنی غیرخطی برازش شده است.

$$V_{OC}(SOC) = \alpha_0 + \alpha_1 SOC \quad (2-31)$$

خطوط زیر در کد، پارامترهای این منحنی خطی را نشان می دهد.

```
%%***** Linearization of Voc *****
qq = [3.6160    0.4218];
a01=qq(1); a11=qq(2);
y0 = a01;
```

ماتریس های فضای حالت مدل زمان پیوسته باتری به همراه بهره سوئیچینگ h در زیر آمده است:

```
%%***** Battery SS Matrices *****
A = [-1/(Rsd*Cb), 0, 0, 0; 0, -1/(R1*C1), 0, 0; -
a11/(Rsd*Cb)+a11/(R2*C2), 1/(R2*C2) - 1/(R1*C1), -1/(R2*C2), 0; 0, 0, 0, -
1/(R2*C2)];
```



```

B = [1/Cb; 1/C1; R0/(R2*C2)+a11/(Cb)+1/C1+1/C2; 1/C2];
T = [0 0 1 0]'; %abs(B)*100
C = [0 0 1 0];
h0 = 0.1;
h = 2*h0;

```

در کد زیر، نحوه طراحی بهره رؤیت‌گر با استفاده از جایابی قطب آمده است. متغیرهای C_d و A_d ماتریس‌های فضای حالت مدل گسسته باتری در رابطه (۲-۲۰) بوده و بردار p قطب‌های مطلوب رؤیت‌گر می‌باشد. متغیر W نیز در رابطه (۲-۲۲) استفاده می‌شود.

```

%%***** Observer Gain Design *****
Ad = A*dt+eye(4);
Cd = C;
p = exp(dt*[-0.001 -0.001 -0.001]);
H = acker(Ad(1:3,1:3)', Cd(1:3)', p)';
H = [H; 0];
W = 1;

```

بعد از تعریف مقادیر ثابت، جریان و ولتاژ ترمینال هر سلول به دست می‌آید. سپس معادلات فضای حالت گسسته (رابطه ۲-۸) برای هر سلول پیاده‌سازی می‌شود تا مقادیر جدید متغیرهای حالت و ولتاژ مدار باز هر سلول به دست آید. این مقادیر برای محاسبه ولتاژ سلول در زمان نمونه‌برداری بعدی استفاده می‌شوند.

```

%%***** Current & Voltage of Cells *****
I1=Ich+Iba11;
I2=Ich+Iba12;
I3=Ich+Iba13;
I4=Ich+Iba14;
V1=Voc1+V11+V12+R0*I1;
V2=Voc2+V21+V22+R0*I2;
V3=Voc3+V31+V32+R0*I3;
V4=Voc4+V41+V42+R0*I4;

%%***** Battery states update *****
%%*****first states update
SOC1_new = (1-dt/(Rsd*Cb))*SOC1+I1*dt/Cb;
V11_new = (1-dt/(R1*C1))*V11+I1*dt/C1;
V12_new = (1-dt/(R2*C2))*V12+I1*dt/C2;
Voc1=a0+a1*SOC1_new+a2*SOC1_new^2+a3*SOC1_new^3+a4*exp(-b*SOC1_new);

%%*****second battery states update
SOC2_new = (1-dt/(Rsd*Cb))*SOC2+I2*dt/Cb;
V21_new = (1-dt/(R1*C1))*V21+I2*dt/C1;
V22_new = (1-dt/(R2*C2))*V22+I2*dt/C2;
Voc2=a0+a1*SOC2_new+a2*SOC2_new^2+a3*SOC2_new^3+a4*exp(-b*SOC2_new);

%%*****third battery states update
SOC3_new = (1-dt/(Rsd*Cb))*SOC3+I3*dt/Cb;
V31_new = (1-dt/(R1*C1))*V31+I3*dt/C1;

```

```
V32_new = (1-dt/(R2*C2))*V32+I3*dt/C2;
Voc3=a0+a1*SOC3_new+a2*SOC3_new^2+a3*SOC3_new^3+a4*exp(-b*SOC3_new);
%****forth battery states update
SOC4_new = (1-dt/(Rsd*Cb))*SOC4+I4*dt/Cb;
V41_new = (1-dt/(R1*C1))*V41+I4*dt/C1;
V42_new = (1-dt/(R2*C2))*V42+I4*dt/C2;
Voc4=a0+a1*SOC4_new+a2*SOC4_new^2+a3*SOC4_new^3+a4*exp(-b*SOC4_new);
```

در ادامه، معادلات SMO برای هر سلول به صورت کد زیر شبیه‌سازی می‌شود. در کد زیر، متغیرهای رویتگر با حرف h همراه نام متغیر مشخص شده‌اند. متغیرهای V_{hi1} , V_{hi2} , V_{hi} و V_{hi} به $\{i=1,2,3,4\}$ ترتیب تخمین SOC، تخمین ولتاژ شبکه RC اول، تخمین ولتاژ شبکه RC دوم و تخمین ولتاژ ترمینال در مدل الکتریکی سلول نام می‌باشند. بردار X_{hi} تخمین متغیرهای حالت سلول نام می‌باشد. متغیر e_y خطای تخمین خروجی می‌باشد و سیگنال v_{ey} تابع سوئیچینگ ناپیوسته در رابطه (۲-۲۹) است که برای پیاده‌سازی آن از تابع تانژانت معکوس (atan) که نسبت به تابع علامت (sign) یکنواخت‌تر است استفاده شده است. مقادیر جدید X_{hi} نیز بر مبنای معادله رویتگر از رابطه (۲-۲۹) به دست می‌آید.

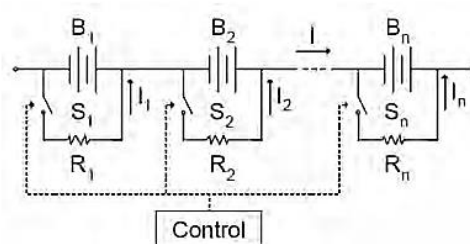
```
%***** SMO Equations *****
%****first battery states update
Xh1 = [SOCh1;Vh11;Vh1;Vh12];
yh1 = C*Xh1+y0;
ey=V1-yh1; v_ey = 2/pi*atan(W*ey); % W*ey/(abs(W*ey)+0.1);
Xh1 = (eye(4)+dt*A)*Xh1+dt*B*I1+dt*H*ey+dt*h*T*v_ey;
SOCh1_new = Xh1(1); Vh11_new = Xh1(2); Vh1_new = Xh1(3); Vh12_new = Xh1(4);
%****second battery states update
Xh2 = [SOCh2;Vh21;Vh2;Vh22];
yh2 = C*Xh2+y0;
ey=V2-yh2; v_ey = 2/pi*atan(W*ey); % W*ey/(abs(W*ey)+0.1);
Xh2 = (eye(4)+dt*A)*Xh2+dt*B*I2+dt*H*ey+dt*h*T*v_ey;
SOCh2_new = Xh2(1); Vh21_new = Xh2(2); Vh2_new = Xh2(3); Vh22_new = Xh2(4);
%****third battery states update
Xh3 = [SOCh3;Vh31;Vh3;Vh32];
yh3 = C*Xh3+y0;
ey=V3-yh3; v_ey = 2/pi*atan(W*ey); % W*ey/(abs(W*ey)+0.1);
Xh3 = (eye(4)+dt*A)*Xh3+dt*B*I3+dt*H*ey+dt*h*T*v_ey;
SOCh3_new = Xh3(1); Vh31_new = Xh3(2); Vh3_new = Xh3(3); Vh32_new = Xh3(4);
%****forth battery states update
Xh4 = [SOCh4;Vh41;Vh4;Vh42];
yh4 = C*Xh4+y0;
ey=V4-yh4; v_ey = 2/pi*atan(W*ey); % W*ey/(abs(W*ey)+0.1);
Xh4 = (eye(4)+dt*A)*Xh4+dt*B*I4+dt*H*ey+dt*h*T*v_ey;
SOCh4_new = Xh4(1); Vh41_new = Xh4(2); Vh4_new = Xh4(3); Vh42_new = Xh4(4);
```

در ادامه این کد، جریان شارژ جدید باتری و نیز فرایند دومرحله‌ای شارژ باتری (CC-CV) پیاده‌سازی می‌شود که عیناً مشابه کد مربوط به الگوریتم EKF است و از آوردن آن در اینجا خودداری می‌شود.

۲-۳- شبیه‌سازی یکسان‌سازی شارژ غیرفعال در MATLAB

در روش‌های غیر فعال یکسان‌سازی شارژ، به هر سلول یک مقاومت اختصاص داده می‌شود که به صورت موازی با آن قرار می‌گیرد (شکل ۲-۲). در این روش، ولتاژ (و یا شارژ) کلیه سلول‌های استک در هر لحظه به سیستم کنترل اعلام شده و سیستم کنترل، فرمان وصل را به کلید متصل به سلول با بیشترین ولتاژ (شارژ) ارسال می‌نماید [۶]. شارژ اضافی سلول با ولتاژ بالاتر تلف می‌شود تا ولتاژ همه سلول‌های استک به یک سطح برسد. مقدار جریانی که به عناصر شنت می‌رسد با ولتاژ سلول متناسب است؛ بنابراین هر چه ولتاژ سلول افزایش یابد مقدار جریانی که به شنت فرستاده می‌شود بیشتر می‌شود. سرعت یکسان‌سازی با انتخاب مقاومت شانت مناسب قابل تنظیم می‌باشد. معایب عمده‌ی این روش به شرح زیر هستند:

- عمل یکسان‌سازی تنها در هنگام شارژ باتری‌ها قابل انجام است.
- بخش زیادی از انرژی رسیده به استک به صورت تلفات در می‌آید و در نتیجه بازدهی این روش یکسان‌سازی نیز مناسب نمی‌باشد.



شکل ۲-۲- مدار یکسان‌ساز شارژ غیرفعال

در ادامه، کدهای نوشته شده در محیط Mfile نرم‌افزار MATLAB برای روش یکسان‌سازی شارژ غیرفعال با استفاده از هر دو روش SMO و EKF که بر روی چهار باتری سری اعمال شده است تشریح خواهد شد.

۲-۳-۱- شبیه‌سازی یکسان‌سازی غیرفعال مبتنی بر تخمین سطح شارژ با استفاده از SMO

در این بخش، کد مربوط به شبیه‌سازی یکسان‌سازی شارژ غیرفعال که در آن از رویکرد SMO برای تخمین SOC استفاده شده است تشریح خواهد شد. در ابتدای این کد، برخی از متغیرهای جهانی^۱ تعریف شده‌اند تا در همه جای برنامه امکان دسترسی به مقدار آن وجود داشته باشد. این متغیرها شامل زمان نمونه‌برداری، پارامترهای مدل باتری، ولتاژ مدار باز هر سلول، ولتاژ دو سر شبکه‌های RC و تخمین این ولتاژها برای هر سلول است.

```
global dt Cb Rsd R0 R1 R2 C1 C2 a0 a1 a2 a3 a4 b y0
global Voc1 Voc2 Voc3 Voc4 V11 V12 V21 V22 V31 V32 V41 V42 Vh11 Vh12 Vh21 Vh22
Vh31 Vh32 Vh41 Vh42
```

¹ global

مقادیر مقاومت شنت در مدار یکسان ساز برابر $R_{bal}=20$ و زمان نمونه برداری برابر $dt=0.5$ انتخاب شده است. همچنین برای کاهش فرکانس سوئیچینگ، یک باند هیستریزس بر روی اختلاف SOC بین سلول‌ها لحاظ شده است که در اینجا مقدار $M_{hyst}=0.005$ انتخاب شده است. این مقدار بر اساس یک مصالحه بین سرعت یکسان سازی و فرکانس سوئیچینگ انتخاب می‌شود.

$dt = 0.5;$
 $R_{bal} = 20;$
 $M_{hyst} = 0.005;$

در ادامه کد، پارامترهای مدل الکتریکی باتری آورده شده است که مقادیر آن در جدول ۲-۲ آمده است. در این جدول، پارامترهای α_i , β ضرایب رابطه VOC بر حسب SOC می‌باشند.

جدول ۲-۲- پارامترهای مدل باتری و مقاومت شنت مدار یکسان ساز

	β	α_4	α_3	α_2	α_1	α_0	پارامتر
	35	-1.031	0.3201	-0.1178	0.2156	3.685	مقدار
C_2	C_1	C_b	R_2	R_1	R_0	R_{sd}	پارامتر
4475	703.6	3060	0.04984	0.04669	0.07446	1e+15	مقدار

خطوط زیر از کد، مربوط به مقداردهی اولیه متغیرهای حالت مربوط به مدل هر سلول، متغیرهای حالت رؤیتگر هر سلول، جریان شارژر باتری (I_{ch})، جریان بالانس هر سلول ($I_{bal,i}$) که جریان گذرنده از مقاومت شنت است، جریان گذرنده از هر سلول (I_i) و ولتاژ ترمینال هر سلول (V_i) می‌باشد. متغیرهای $V_{i1}, V_{i2} \{i = 1, 2, 3, 4\}$ بیانگر ولتاژهای دو سر شبکه‌های RC در شکل ۱-۲ است. متغیرهای رؤیتگر با یک حرف h همراه نام متغیر مشخص شده‌اند. لازم به ذکر است که ظرفیت سلول مورد مطالعه برابر 0.85 Ah است و لذا جریان شارژی کمتر از نرخ 1C به باتری اعمال می‌شود. جریان کل گذرنده از هر سلول برابر با جریان شارژر بعلاوه جریان بالانس آن سلول است.

```
***** initial states of batteries
SOC1(1)=.6;
SOC2(1)=.4;
SOC3(1)=.35;
SOC4(1)=.3;
V11=0; V12=0; V21=0; V22=0; V31=0; V32=0; V41=0; V42=0;
Voc1=a0+a1*SOC1+a2*SOC1^2+a3*SOC1^3+a4*exp(-b*SOC1);
Voc2=a0+a1*SOC2+a2*SOC2^2+a3*SOC2^3+a4*exp(-b*SOC2);
Voc3=a0+a1*SOC3+a2*SOC3^2+a3*SOC3^3+a4*exp(-b*SOC3);
Voc4=a0+a1*SOC4+a2*SOC4^2+a3*SOC4^3+a4*exp(-b*SOC4);

Vh11=0; Vh12=0; Vh21=0; Vh22=0; Vh31=0; Vh32=0; Vh41=0; Vh42=0;
Vh1(1)=0; Vh2(1)=0; Vh3(1)=0; Vh4(1)=0;
SOCh1(1)=SOC1(1); SOCh2(1)=SOC2(1); SOCh3(1)=SOC3(1); SOCh4(1)=SOC4(1);

Ich(1)=0.75;
Ibal1(1)=0;
Ibal2(1)=0;
Ibal3(1)=0;
```

```

Iba14(1)=0;
I1(1)=Ich(1)+Iba11(1);
I2(1)=Ich(1)+Iba12(1);
I3(1)=Ich(1)+Iba13(1);
I4(1)=Ich(1)+Iba14(1);

V1(1)=Voc1+V11+V12+R0*I1;
V2(1)=Voc2+V21+V22+R0*I2;
V3(1)=Voc3+V31+V32+R0*I3;
V4(1)=Voc4+V41+V42+R0*I4;

```

برنامه مربوط به یکسان‌سازی شارژ سلول‌ها آنقدر ادامه می‌یابد تا SOC سلول‌ها در محدوده مشخصی نسبت به هم قرار گیرد. برای این منظور، از حلقه while در MATLAB استفاده شده است. در خطوط ابتدایی این حلقه، کدی برای ترسیم مناسب شکل‌های ولتاژ و جریان ترمینال سلول‌ها نوشته شده است که متغیرهای استفاده شده در آن (V_{i_fig} و I_{i_fig}) در داخل برنامه اصلی استفاده نمی‌شود.

در این حلقه، ابتدا یک تابع فراخوانی می‌شود که در آن جریان و ولتاژ ترمینال هر سلول محاسبه (اندازه‌گیری) شده و به تخمین‌گر SOC داده می‌شود. تخمین‌گر هر سلول با داشتن این دو مقدار، SOC جدید آن سلول را تخمین می‌زند، که این تخمین‌گر SMO یا EKF است. همچنین جریان شارژ جدید باتری تعیین می‌شود.

خطوط زیر در کد، مربوط به فراخوانی تابع States_Ich_Update_SOCbalancing_SMO برای حالتی است که از SMO استفاده شده است. ورودی‌های این تابع، مقادیر فعلی جریان شارژ، جریان بالانس سلول‌ها، SOC سلول‌ها، SOC تخمین زده شده سلول‌ها و ولتاژ تخمین زده شده سلول‌ها می‌باشد. خروجی‌های این تابع نیز جریان و ولتاژ ترمینال فعلی سلول‌ها و مقادیر جدید جریان شارژ، SOC سلول‌ها، SOC تخمین زده شده سلول‌ها و ولتاژ تخمین زده شده سلول‌ها می‌باشد. جزئیات این تابع در فصل دوم آمده است.

```

[V1,V2,V3,V4,I1,I2,I3,I4,Ich_new,SOC1_new,SOC2_new,SOC3_new,SOC4_new,Vh1_new,Vh2_new,Vh3_new,Vh4_new,SOCh1_new,SOCh2_new,SOCh3_new,SOCh4_new]=
States_Ich_Update_SOCbalancing_SMO(Vh1,Vh2,Vh3,Vh4,SOCh1,SOCh2,SOCh3,SOCh4,SOC1,
SOC2,SOC3,SOC4,Iba11,Iba12,Iba13,Iba14,Ich)

```

بعد از اینکه SOC سلول‌ها تخمین زده شد، مقادیر آن‌ها از بزرگ به کوچک مرتب می‌شود.

```

%****SORTING THE CELLS ACCORDING TO THEIR SOC
SOC_matrix=[SOCh1(k+1),SOCh2(k+1),SOCh3(k+1),SOCh4(k+1)];
[A,B]=sort(SOC_matrix);

```

حال از روی این مقادیر SOC تصمیم گرفته می‌شود که آیا نیاز به انجام بالانس می‌باشد یا خیر. این تصمیم‌گیری معمولاً از روی اختلاف بین بزرگترین و کوچکترین مقادیر SOC سلول‌ها صورت می‌گیرد که حداکثر اختلاف به اندازه ۱٪ در نظر گرفته شده است. دو حالت وجود دارد که در ادامه آورده شده است.

الف- اختلاف بزرگتر از ۱٪ بوده و نیاز به انجام بالانس می‌باشد:

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%SHOULD WE START CELL EQUALIZATION OR NOT
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if (SOC_matrix(B(4))-SOC_matrix(B(1)))>.01 % max(SOCi)-min(SOCi)>.01

```

در این حالت ابتدا باید سلول با بالاترین SOC انتخاب شود و مقاومت شنت متصل به آن از طریق سوئیچ به دو سر آن وصل شود تا شارژ اضافه موجود در این سلول تخلیه شود. فرض کنید سلول‌های ۱، ۲، ۳ و ۴ به ترتیب دارای SOC از بالاترین مقدار تا پایین‌ترین مقدار می‌باشند. سلول شماره ۱ که دارای بالاترین SOC است، سوئیچ آن وصل شده و جریان بالانس گذرنده از مقاومت شنت آن از قانون اهم به راحتی به دست می‌آید. لازم به ذکر است که جریان شارژ باتری به عنوان جهت مثبت جریان در نظر گرفته شده است و لذا علامت جریان بالانس منفی می‌باشد. با داشتن مقادیر فعلی جریان بالانس و مقادیر فعلی SOC واقعی و مقادیر تخمین زده شده SOC و ولتاژ سلول‌ها (که از فراخوانی قبلی تابع به دست آمده است)، بار دیگر تابع فراخوانی می‌شود و خروجی‌های تابع که جریان و ولتاژ ترمینال فعلی سلول‌ها و مقادیر جدید جریان شارژ، SOC سلول‌ها، SOC تخمین زده شده سلول‌ها و ولتاژ تخمین زده شده سلول‌ها می‌باشد محاسبه می‌شوند. تخلیه سلول شماره ۱ که دارای سطح بالاتری SOC است تا جایی ادامه می‌یابد که SOC آن به اندازه یک مقدار مشخص از SOC سلول شماره ۲ (که SOC آن کمتر از سلول ۱ و بیشتر از بقیه است) پایین‌تر رود. در واقع یک باند هیستریزس در اینجا لحاظ کرده‌ایم تا فرکانس سوئیچینگ کاهش یابد که در اینجا 0.005 انتخاب شده است.

```

if SOC_matrix(B(4))==SOCh1(k+1)
    while (SOC_matrix(B(4))-SOC_matrix(B(3)))> -Mhyst
        Ibal1(k+1)=-V1(k)/Rbal;
        Ibal2(k+1)=0;
        Ibal3(k+1)=0;
        Ibal4(k+1)=0;
        k=k+1;

[V1(k),V2(k),V3(k),V4(k),I1(k),I2(k),I3(k),I4(k),Ich(k+1),SOC1(k+1),SOC2(k+1),SOC3(k+1),
SOC4(k+1),Vh1(k+1),Vh2(k+1),Vh3(k+1),Vh4(k+1),SOCh1(k+1),SOCh2(k+1),
SOCh3(k+1),SOCh4(k+1)] = ...
States_Ich_Update_SOCbalancing_SMO(Vh1(k),Vh2(k),Vh3(k),Vh4(k),SOCh1(k),SOCh2(k),SOCh3(k),
SOCh4(k),SOC1(k),SOC2(k),SOC3(k),SOC4(k),Ibal1(k),Ibal2(k),Ibal3(k),Ibal4(k),Ich(k)));

    SOC_matrix=[SOCh1(k+1),SOCh2(k+1),SOCh3(k+1),SOCh4(k+1)];

    End
end

```

به همین ترتیب اگر سلول‌های ۲ و ۳ و ۴ دارای بالاترین SOC باشند نیز مشابه فوق عمل می‌شود.

ب- اختلاف کوچکتر از ۱٪ بوده و نیاز به انجام بالانس نمی‌باشد:

در این حالت، جریان بالانس همه سلول‌ها صفر بوده و هیچ سوئیچی وصل نیست. فقط جریان و ولتاژ ترمینال هر سلول اندازه‌گیری می‌شود. جریان گذرنده از همه سلول‌ها برابر جریان شارژ می‌باشد.

```

else
    Ibal1(k+1)=0;
    Ibal2(k+1)=0;
    Ibal3(k+1)=0;
    Ibal4(k+1)=0;

```

```

I1(k+1)=Ich(k+1);
I2(k+1)=Ich(k+1);
I3(k+1)=Ich(k+1);
I4(k+1)=Ich(k+1);
k=k+1;
V1(k)=Voc1+V11+V12+R0*I1(k);
V2(k)=Voc2+V21+V22+R0*I2(k);
V3(k)=Voc3+V31+V32+R0*I3(k);
V4(k)=Voc4+V41+V42+R0*I4(k);
end

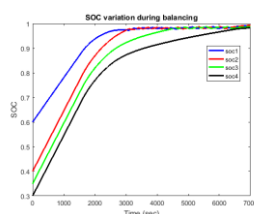
```

این کار آنقدر ادامه می‌یابد تا پک باتری کاملاً شارژ شود که شرط شارژ کامل، رسیدن جریان شارژ به یک حد کمینه (در اینجا 1mA) است که در داخل تابع آورده شده است.

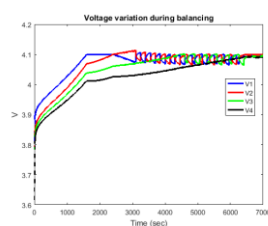
در ادامه این برنامه، به ولتاژ ترمینال تخمین زده شده، مقدار ثابتی (عرض از مبدأ) که از خطی‌سازی منحنی $V_{oc}(SOC)$ به دست می‌آید اضافه می‌شود. به منظور خطی کردن معادلات رؤیت‌گر، این مقدار نادیده گرفته شده بود.

```
Vh1 = Vh1+y0; Vh2 = Vh2+y0; Vh3 = Vh3+y0; Vh4 = Vh4+y0;
```

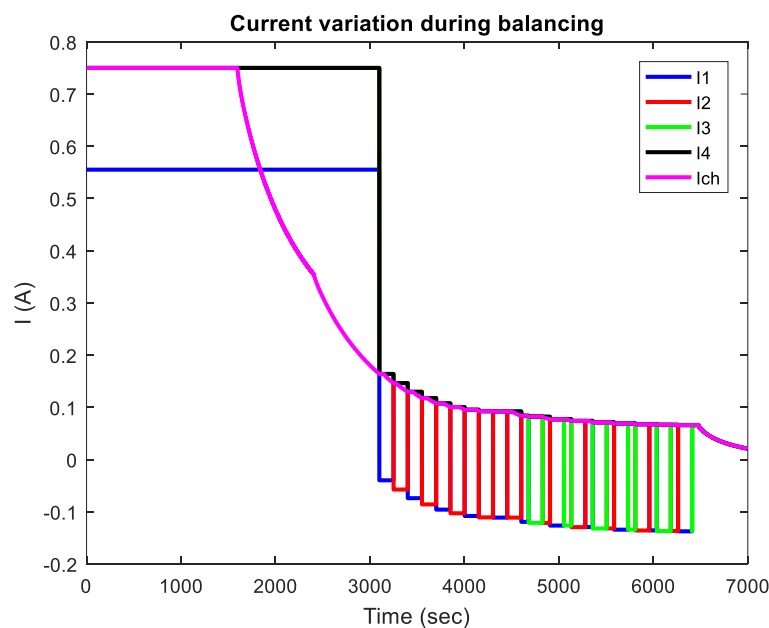
نتایج اجرای این کد در شکل‌های زیر آمده است. مشاهده می‌کنید که SOC سلول‌ها در طول فرایند یکسان‌سازی به هم نزدیک شده و در نهایت با فاصله حدود ۱٪ از هم قرار گرفته‌اند. پس از این زمان دیگر سوئیچی بین سلول‌ها انجام نشده است. در طول این فرایند، ولتاژها نیز به هم نزدیک شده‌اند. جریان سلول‌ها نیز در ابتدا ثابت بوده (مرحله CC) و سپس کاهش یافته (مرحله CV) و پس از اینکه فاصله SOC سلول‌ها کمتر از ۱٪ شده است، دیگر سوئیچینگ انجام نشده است و شارژ سلول‌ها تا رسیدن جریان به کمتر از 1mA ادامه پیدا کرده است.



شکل ۳-۲- SOC سلول‌ها در طی یکسان‌سازی غیرفعال با روش SMO



شکل ۲-۴- ولتاژ سلول‌ها در طی یکسان‌سازی غیرفعال با روش SMO



شکل ۲-۵- جریان سلول‌ها در طی یکسان‌سازی غیرفعال با روش SMO

۲-۳-۲- شبیه‌سازی یکسان‌سازی غیرفعال مبتنی بر تخمین سطح شارژ با استفاده از EKF

در این بخش، کد مربوط به شبیه‌سازی یکسان‌سازی شارژ غیرفعال که در آن از تخمینگر EKF برای تخمین SOC استفاده شده است تشریح خواهد شد. در حالتی که از EKF به جای SMO برای تخمین SOC استفاده می‌شود کدی مشابه کد فوق نوشته شده است و فقط چند تفاوت وجود دارد که در زیر اشاره شده است.

در ابتدای این کد، علاوه بر متغیرها جهانی که در کد قبلی تعریف شده‌اند، کوواریانس خطا در تخمینگر هر سلول نیز به عنوان متغیرهای جهانی تعریف شده است.

```
global dt Cb Rsd R0 R1 R2 C1 C2 a0 a1 a2 a3 a4 b
global Voc1 Voc2 Voc3 Voc4 V11 V12 V21 V22 V31 V32 V41 V42 Vh11 Vh12 Vh21 Vh22 Vh31
Vh32 Vh41 Vh42 P1 P2 P3 P4
```

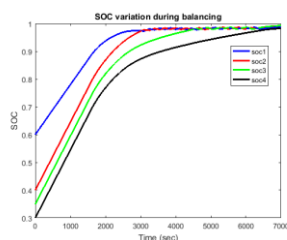
در بخش مقداردهی اولیه متغیرها، علاوه بر مقداردهی متغیرهای بخش قبل، کوواریانس‌های خطا نیز مقداردهی می‌شوند.

```
P1 = 1e-5*eye(3);
P2 = 1e-5*eye(3);
P3 = 1e-5*eye(3);
P4 = 1e-5*eye(3);
```

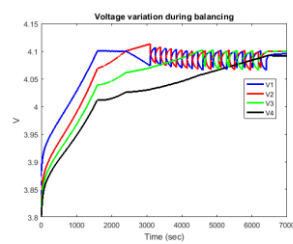
به جای تابع مربوط به SMO، تابع States_Ich_Update_SOCbalancing_EKF در هر بار فراخوانی می‌شود. ورودی‌های این تابع، مقادیر فعلی جریان شارژ، جریان بالانس سلول‌ها، SOC سلول‌ها، تخمین زده شده سلول‌ها و ولتاژ تخمین زده شده سلول‌ها می‌باشد. خروجی‌های این تابع نیز جریان و ولتاژ ترمینال فعلی سلول‌ها و مقادیر جدید جریان شارژ، SOC سلول‌ها، SOC تخمین زده شده سلول‌ها و ولتاژ تخمین زده شده سلول‌ها می‌باشد. جزئیات این تابع در فصل دوم آمده است.

```
function[V1,V2,V3,V4,I1,I2,I3,I4,Ich_new,SOC1_new,SOC2_new,SOC3_new,SOC4_new,Vh1,Vh2,Vh3,Vh4,SOC1_new,SOC2_new,SOC3_new,SOC4_new] = ...
States_Ich_Update_SOCbalancing_EKF (SOC1,SOC2,SOC3,SOC4,Ibal1,Ibal2,Ibal3,Ibal4,Ich)
```

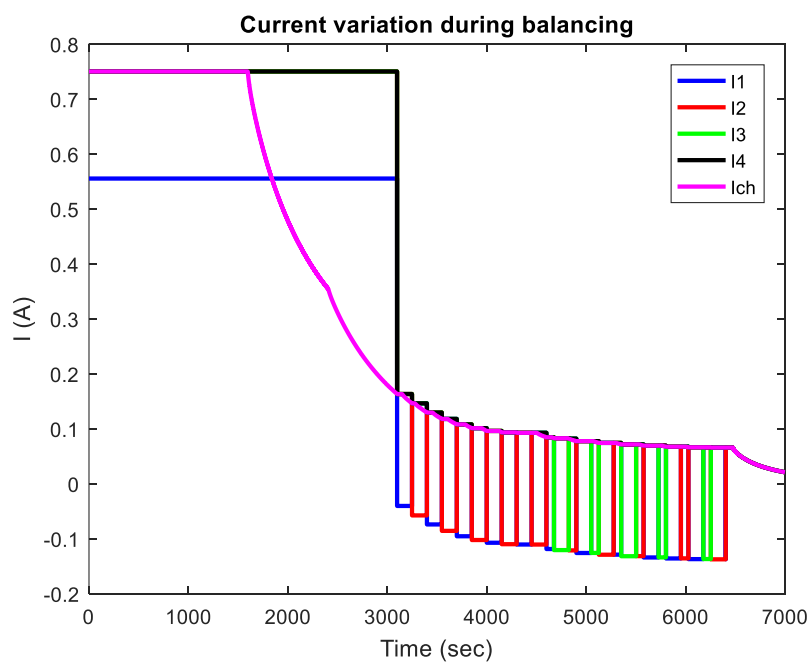
نتایج اجرای این کد در شکل‌های زیر آمده است.



شکل ۶-۲ SOC سلول‌ها در طی یکسان‌سازی غیرفعال با روش EKF



شکل ۲-۷- ولتاژ سلول‌ها در طی یکسان‌سازی غیرفعال با روش EKF



شکل ۲-۸- جریان سلول‌ها در طی یکسان‌سازی غیرفعال با روش EKF

۲-۴- شبیه‌سازی یکسان‌سازی شارژ فعال در MATLAB

در یکسان‌سازهای فعال، انرژی اضافی سلول‌ها به جای اینکه تلف شود، به وسیله‌ی عناصر فعالی نظیر خازن، مبدل‌های DC به DC ایزوله و یا غیر ایزوله و یا ترکیبی از آن‌ها به سلول‌های با شارژ کمتر منتقل گشته و به این ترتیب باعث بالا رفتن بازدهی این روش‌ها نسبت به روش‌های یکسان‌سازی غیرفعال خواهد شد. در این پروژه از روش یکسان‌سازی شارژ فعال با مبدل مبتنی بر ترانسفرمر ایزوله (مبدل فلای‌بک^۱ مرکزی) استفاده شده است. عنصر اصلی این مبدل‌های فلای‌بک، یک هسته با دو و یا چند سیم‌پیچ است. شکل ۹-۲ نمایش‌دهنده‌ی مدار یکسان‌ساز مبتنی بر مبدل‌های فلای‌بک چندسیم‌پیچه می‌باشند که در این پروژه مورد استفاده قرار گرفته است [۸]. در ادامه به بررسی حالات مختلف عملکردی مدار می‌پردازیم. فلش‌های مختلف، جهت جریان را در هر بخش نشان می‌دهد و بخش فعال مدار پررنگ‌تر شده است.

الف- مرحله‌ی اول (t_0 تا t_1)

نخست سیستم کنترل، تمام ولتاژهای اندازه‌گیری شده را بررسی کرده و سلول با بیشترین ولتاژ را انتخاب می‌نماید. در صورتیکه سلول انتخاب شده B1 باشد، کلید S1 بسته شده و حالت مدار به صورت نشان‌داده شده در شکل ۱۰-۲ در می‌آید.

با توجه به اینکه جریانی از ثانویه‌ی ترانس عبور نمی‌کند، جریان عبوری از سیم‌پیچ متصل به B1 به صورت خطی افزایش می‌یابد. سایر کلیدها در این حالت قطع بوده و بنابراین جریانی از هیچ یک از سلول‌های دیگر کشیده نمی‌شود. بنابراین خواهیم داشت:

$$\begin{cases} i_{L1}(t) = \frac{1}{L_m} \int_{t_0}^t V_{B1}(\tau) d\tau \\ i_{bal1}(t) = -i_{L1}(t) \\ i_{L2}(t) = 0 \\ i_{bal2}(t) = 0 \end{cases} \begin{cases} i_{L3}(t) = 0 \\ i_{bal3}(t) = 0 \\ i_{L4}(t) = 0 \\ i_{bal4}(t) = 0 \end{cases} \quad (2-32)$$

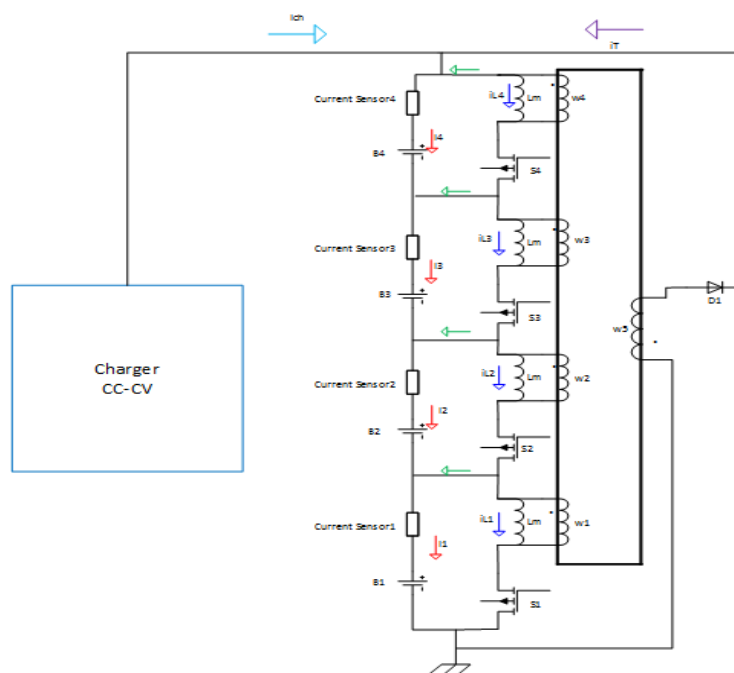
با توجه به روابط فوق، می‌توان جریان گذرنده از هر سلول را به صورت مجموع جریان شارژ و جریان یکسان‌سازی آن در نظر گرفت. همچنین در تمام طول این دوره، جریان عبوری از ثانویه‌ی ترانس برابر با صفر می‌باشد.

$$I_i(t) = I_{ch}(t) + I_{bal_i}(t) \quad (2-33)$$

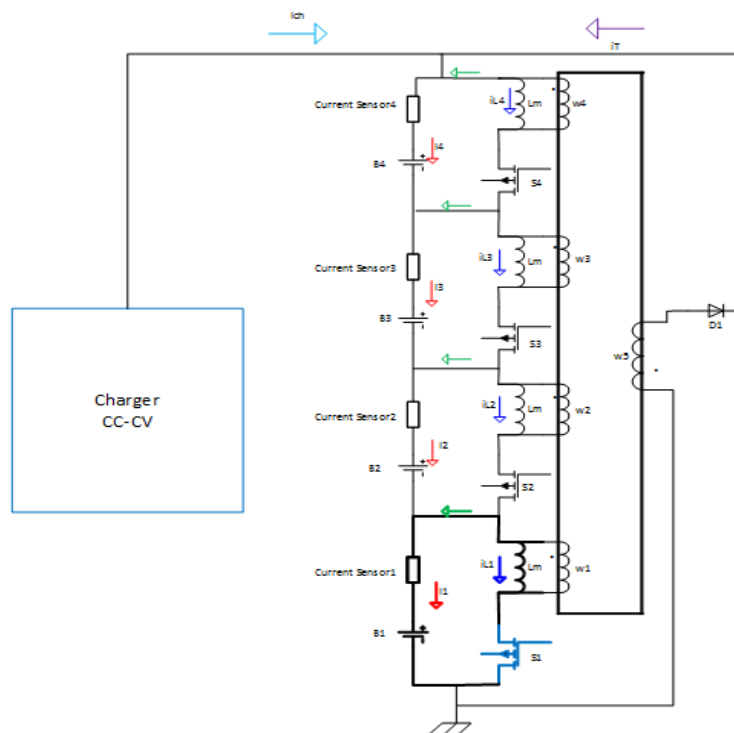
$$i_T(t) = 0 \quad (2-34)$$

با توجه به اینکه در این فاصله زمانی ثانویه‌ی ترانس قطع می‌باشد، بنابراین تمام انرژی خارج شده از سلول B1 در اندوکتانس مغناطیس‌کنندگی ترانس ذخیره شده است.

¹ Flyback

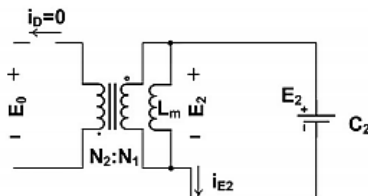


شکل ۹-۲- یکسان ساز با مبدل فلای‌بک مرکزی با سیم‌پیچ‌های ثانویه متعدد



شکل ۱۰-۲- وضعیت مدار یکسان‌ساز در حالتی که کلید S1 روشن می‌باشد

لازم به ذکر است که در مبدل‌های فلای‌بک معمولاً از روش کنترل جریان بیشینه^۱ (Ipeak) استفاده می‌کنند. در این روش، جریان گذرنده از سیم‌پیچ اندازه گرفته می‌شود و با رسیدن این جریان به یک حد بیشینه‌ی از پیش تعریف شده، کلید مربوطه جریان را قطع می‌کند. شکل ۱۱-۲ مدار ساده شده یکسان‌ساز با مبدل فلای‌بک به کار گرفته شده را نمایش می‌دهد [۹].



شکل ۱۱-۲- مدار ساده شده یکسان‌ساز با مبدل فلای‌بک

ب- مرحله‌ی دوم (t_1 تا t_2)

پس از اینکه جریان اولیه‌ی ترانس به مقدار بیشینه (Ipeak) رسید، فرمان قطع به سلول S1 صادر می‌شود و در نتیجه انرژی ذخیره شده در اندوکتانس مغناطیس‌کنندگی (L_m) به ثانویه‌ی ترانس منتقل شده و دیود D1 در ثانویه ترانس هدایت می‌کند. در نتیجه ولتاژ دو سر سیم‌پیچ ثانویه برابر با ولتاژ کل استک خواهد بود. در این فاصله‌ی زمانی، تغییرات جریان گذرنده از L_m و کل استک به صورت زیر بیان می‌گردد:

$$\begin{cases} i_{L1}(t) = i_{L1}(t_1) - \int_{t_1}^t \frac{1}{n} (V_{B1}(\tau) + V_{B2}(\tau) + V_{B3}(\tau) + V_{B4}(\tau)) d\tau \\ i_{L2}(t) = i_{L3}(t) = i_{L4}(t) = 0 \\ i_{D1}(t) = i_T(t) = \frac{1}{n} i_{L1}(t) \\ i_{bal1}(t) = i_{bal2}(t) = i_{bal3}(t) = i_{bal4}(t) = i_T(t) \end{cases} \quad (2-35)$$

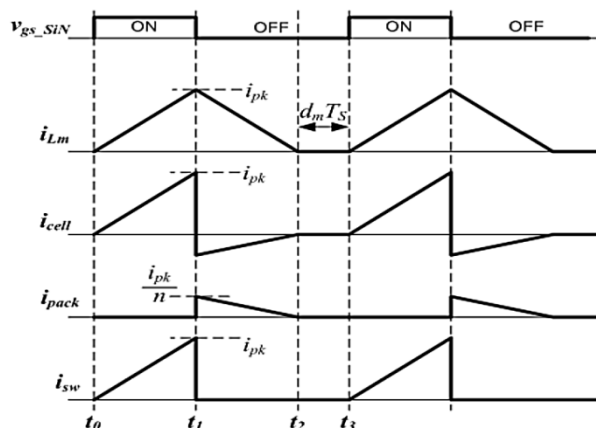
که در آن $n = N_2/N_1$. این مرحله تا زمانی ادامه می‌یابد که جریان برقرار شده در ثانویه‌ی ترانس به صفر برسد.

پ- مرحله سوم (t_2 تا t_3)

این مرحله از لحظه‌ای که جریان ثانویه به صفر رسیده است شروع می‌شود و تا پایان آن هیچ‌یک از کلیدها هدایت نخواهند نمود.

شکل ۱۲-۲ جریان‌ها و ولتاژ کلیدهای مختلف را طی مراحل سه‌گانه‌ی شرح داده شده در فوق نمایش می‌دهد [۹]. در این شکل، ولتاژ بین گیت و سورس کلید متصل به سلول با بیشترین ولتاژ بوده و i_{Lm} ، i_{sw} و i_{pack} به ترتیب نشان‌دهنده‌ی جریان گذرنده از اندوکتانس مغناطیس‌کنندگی، جریان سلول متصل به کلید، جریان یکسان‌سازی کل استک و جریان گذرنده از کلید می‌باشند.

¹ Peak current mode control



شکل ۱۲-۲- وضعیت ولتاژها و جریان ها مختلف مدار یکسان ساز طی مراحل مختلف

در ادامه، کدهای نوشته شده در محیط Mfile نرم افزار MATLAB برای روش یکسان سازی شارژ فعال مبتنی بر مبدل فلای بک مرکزی با استفاده از هر دو روش SMO و EKF که بر روی چهار باتری سری اعمال شده است تشریح خواهد شد.

۱-۴-۲- شبیه سازی یکسان سازی فعال مبتنی بر مبدل فلای بک با استفاده از تخمینگر SMO

در این بخش، کد مربوط به شبیه سازی یکسان سازی شارژ فعال که در آن از رؤیتگر SMO برای تخمین SOC استفاده شده است تشریح خواهد شد. مشابه کد مربوط به یکسان سازی غیرفعال، در ابتدای این کد، برخی از متغیرها جهانی تعریف شده و پارامترهای مدل الکتریکی باتری آورده شده است.

در ادامه این کد، مشخصات مدار یکسان ساز شامل اندوکتانس مغناطیس کنندگی (L_m) و بیشینه جریان (I_{peak}) گذرنده از این اندوکتانس انتخاب شده اند. مقدار اندوکتانس توسط طراح طراحی می شود و مقدار جریان بیشینه نیز از روی ظرفیت سلول انتخاب می شود. همچنین نسبت تبدیل ترانس برابر $n=4$ در نظر گرفته شده است.

```
%cell equalizing circuit characteristics
Imax1=.8;
Imax2=Imax1/4;
L1=10;
```

کد زیر مربوط به مقداردهی اولیه متغیرها است که علاوه بر متغیرهای استفاده شده در یکسان سازی غیرفعال، متغیرهای I_T و I_{Li} نیز اضافه شده است. در اینجا نیز یک مقدار هیستریزیس روی اختلاف SOC سلول ها به منظور کاهش سوئیچینگ در نظر گرفته شده است که برابر 0.005 می باشد.

```
***** initial states of batteries
SOC1(1)=.9;
SOC2(1)=.8;
SOC3(1)=.6;
SOC4(1)=.5;
V11=0; V12=0; V21=0; V22=0; V31=0; V32=0; V41=0; V42=0;
Voc1=3.9;
Voc2=3.83;
Voc3=3.82;
Voc4=3.78;
```

```
Vh11=0; Vh12=0; Vh21=0; Vh22=0; Vh31=0; Vh32=0; Vh41=0; Vh42=0;
Vh1(1)=0; Vh2(1)=0; Vh3(1)=0; Vh4(1)=0;
SOCh1(1)=SOC1(1); SOCh2(1)=SOC2(1); SOCh3(1)=SOC3(1); SOCh4(1)=SOC4(1);

Ich(1)=0.55;
Ibal1(1)=0;
Ibal2(1)=0;
Ibal3(1)=0;
Ibal4(1)=0;
I1(1)=Ich(1)+Ibal1(1);
I2(1)=Ich(1)+Ibal2(1);
I3(1)=Ich(1)+Ibal3(1);
I4(1)=Ich(1)+Ibal4(1);
iT(1)=0;
iL1(1)=0;
iL2(1)=0;
iL3(1)=0;
iL4(1)=0;

V1(1)=Voc1;
V2(1)=Voc2;
V3(1)=Voc3;
V4(1)=Voc4;
```

برنامه مربوط به یکسان‌سازی شارژ سلول‌ها آنقدر ادامه می‌یابد تا SOC سلول‌ها در محدوده مشخصی نسبت به هم قرار گیرد. برای این منظور، از حلقه while در MATLAB استفاده شده است. در خطوط ابتدایی این حلقه، کدی برای ترسیم مناسب شکل‌های ولتاژ و جریان ترمینال سلول‌ها نوشته شده است که متغیرهای استفاده شده در آن (V_{i_fig} و I_{i_fig}) در داخل برنامه اصلی استفاده نمی‌شود. این کد، برای یکنواخت کردن منحنی ولتاژ و جریان در زمان ترسیم است و برای این منظور، ولتاژ و جریان را در بازه زمانی که یک سلول انتخاب می‌شود برابر با ولتاژ و جریان ابتدای آن بازه قرار می‌دهد.

```
if k~=1
    V1_fig(1)=V1(2);
    V2_fig(1)=V2(2);
    V3_fig(1)=V3(2);
    V4_fig(1)=V4(2);
    I1_fig(1)=I1(2);
    I2_fig(1)=I2(2);
    I3_fig(1)=I3(2);
    I4_fig(1)=I4(2);

    V1_fig(j1+1:1:k-1)=V1_fig(j1);
    V1_fig(k)=V1(k);
    V2_fig(j1+1:1:k-1)=V2_fig(j1);
    V2_fig(k)=V2(k);
    V3_fig(j1+1:1:k-1)=V3_fig(j1);
    V3_fig(k)=V3(k);
    V4_fig(j1+1:1:k-1)=V4_fig(j1);
    V4_fig(k)=V4(k);
```

```
I1_fig(j1+1:1:k-1) = I1_fig(j1);
I1_fig(k)=I1(k);
I2_fig(j1+1:1:k-1) = I2_fig(j1);
I2_fig(k)=I2(k);
I3_fig(j1+1:1:k-1) = I3_fig(j1);
I3_fig(k)=I3(k);
I4_fig(j1+1:1:k-1) = I4_fig(j1);
I4_fig(k)=I4(k);
sd(j1+1:1:k-1)=sd(j1);
sd(k)=std([V1(k),V2(k),V3(k),V4(k)]);
j1=k;
end
```

سپس تابع States_Ich_Update_SOCbalancing_SMO فراخوانی می‌شود که در آن الگوریتم SMO برای تخمین SOC باتری شبیه‌سازی شده است و همچنین جریان شارژ لحظه بعد تعیین می‌شود. ورودی‌های این تابع، مقادیر فعلی جریان شارژ، جریان بالانس سلول‌ها، SOC سلول‌ها، SOC تخمین زده شده سلول‌ها و ولتاژ تخمین زده شده سلول‌ها می‌باشد. خروجی‌های این تابع نیز جریان و ولتاژ ترمینال فعلی سلول‌ها و مقادیر جدید جریان شارژ، SOC سلول‌ها، SOC تخمین زده شده سلول‌ها و ولتاژ تخمین زده شده سلول‌ها می‌باشد. این تابع در فصل دوم به تفصیل توضیح داده شده است.

```
[V1(k),V2(k),V3(k),V4(k),I1(k),I2(k),I3(k),I4(k),Ich(k+1),SOC1(k+1),SOC2(k+1),
SOC3(k+1),SOC4(k+1),Vh1(k+1),Vh2(k+1),Vh3(k+1),Vh4(k+1),SOCh1(k+1),SOCh2(k+1),
SOCh3(k+1),SOCh4(k+1)] = ...
states_Ich_update_SOCbalancing_SMO(Vh1(k),Vh2(k),Vh3(k),Vh4(k),SOCh1(k),SOCh2(k),SOCh3
(k),SOCh4(k),SOC1(k),SOC2(k),SOC3(k),SOC4(k),Iba11(k),Iba12(k),Iba13(k),Iba14(k),Ich(k
));
```

بعد از اینکه SOC سلول‌ها تخمین زده شد، مقادیر آن‌ها از بزرگ به کوچک مرتب می‌شود.

```
*****SORTING THE CELLS ACCORDING TO THEIR SOC
SOC_matrix=[SOCh1(k+1),SOCh2(k+1),SOCh3(k+1),SOCh4(k+1)];
[A,B]=sort(SOC_matrix);
```

حال از روی این مقادیر SOC تصمیم گرفته می‌شود که آیا نیاز به انجام بالانس می‌باشد یا خیر. این تصمیم‌گیری معمولاً از روی اختلاف بین بزرگترین و کوچکترین مقادیر SOC سلول‌ها صورت می‌گیرد که حداکثر اختلاف به اندازه ۱٪ در نظر گرفته شده است. دو حالت وجود دارد که در ادامه آورده شده است.

الف- اختلاف بزرگتر از ۱٪ بوده و نیاز به انجام بالانس می‌باشد:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%SHOULD WE START CELL EQUALIZATION OR NOT
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if (SOC_matrix(B(4))-SOC_matrix(B(1)))>.01
```

در این حالت ابتدا باید سلول با بالاترین SOC انتخاب شود. فرض کنید سلول‌های ۱، ۲، ۳ و ۴ به ترتیب دارای SOC از بالاترین مقدار تا پایین‌ترین مقدار می‌باشند. سلول شماره ۱ که دارای بالاترین SOC باشد سوئیچ آن وصل شده و جریان از طریق سلف مغناطیس‌کنندگی در سمت اولیه ترانس ذخیره می‌شود. این کار تا جایی ادامه می‌یابد که جریان به یک مقدار بیشینه از پیش تعیین شده (I_{max1}) برسد. حال مقادیر جریان بالانس هر سلول محاسبه شده و تابع فراخوانی می‌شود و خروجی‌های تابع که جریان و ولتاژ ترمینال فعلی

سلول‌ها و مقادیر جدید جریان شارژ، SOC سلول‌ها، SOC تخمین زده شده سلول‌ها و ولتاژ تخمین زده شده سلول‌ها می‌باشد محاسبه می‌شوند.

```
while il1(k)<Imax1
    il1(k+1)=il1(k)+V1(k)*dt/L1;
    il2(k+1)=0;
    il3(k+1)=0;
    il4(k+1)=0;
    iT(k+1)=0;
    Ibal1(k+1)=-il1(k+1)+iT(k+1);
    Ibal2(k+1)=0;
    Ibal3(k+1)=0;
    Ibal4(k+1)=0;
    [V1(k+1),V2(k+1),V3(k+1),V4(k+1),I1(k+1),I2(k+1),I3(k+1),I4(k+1),Ich(k+2),SOC1(k+2),SOC2(k+2),SOC3(k+2),SOC4(k+2),Vh1(k+2),Vh2(k+2),Vh3(k+2),Vh4(k+2),SOCh1(k+2),SOCh2(k+2),SOCh3(k+2),SOCh4(k+2)] =
    states_Ich_update_SOCbalancing_SMO
    (Vh1(k+1),Vh2(k+1),Vh3(k+1),Vh4(k+1),SOCh1(k+1),SOCh2(k+1),SOCh3(k+1),SOCh4(k+1),SOC1(k+1),SOC2(k+1),SOC3(k+1),SOC4(k+1),Ibal1(k+1),Ibal2(k+1),Ibal3(k+1),Ibal4(k+1),Ich(k+1));
    k=k+1;
    t(k)=t(k-1)+dt;
end
```

پس از مرحله فوق، انرژی ذخیره شده در اندوکتانس مغناطیس‌کنندگی ترانس به ثانویه منتقل می‌شود. کد زیر، پیاده‌سازی روابط (۳-۴) با انتخاب $n=4$ می‌باشد. خط دوم در این کد برای جلوگیری از منفی شدن جریان است. بعد از هر گام نمونه‌برداری، یک بار تابع فوق‌الذکر فراخوانی شده و SOC تخمین زده شده و جریان شارژ جدید به دست می‌آید. این مرحله تا زمانی که جریان ثانویه به صفر نرسد ادامه می‌یابد.

```
while il1(k)>0
    il1(k+1)=il1(k)-.25*dt*(V1(k)+V2(k)+V3(k)+V4(k))/L1;
    il1(k+1)=il1(k+1)*.5*(1+sign(il1(k+1)));
    il2(k+1)=0;
    il3(k+1)=0;
    il4(k+1)=0;
    iT(k+1)=.25*il1(k+1);
    Ibal1(k+1)=iT(k+1);
    Ibal2(k+1)=iT(k+1);
    Ibal3(k+1)=iT(k+1);
    Ibal4(k+1)=iT(k+1);
    [V1(k+1),V2(k+1),V3(k+1),V4(k+1),I1(k+1),I2(k+1),I3(k+1),I4(k+1),Ich(k+2),SOC1(k+2),SOC2(k+2),SOC3(k+2),SOC4(k+2),Vh1(k+2),Vh2(k+2),Vh3(k+2),Vh4(k+2),SOCh1(k+2),SOCh2(k+2),SOCh3(k+2),SOCh4(k+2)] = ...
    states_Ich_update_SOCbalancing_SMO(Vh1(k+1),Vh2(k+1),Vh3(k+1),Vh4(k+1),SOCh1(k+1),SOCh2(k+1),SOCh3(k+1),SOCh4(k+1),SOC1(k+1),SOC2(k+1),SOC3(k+1),SOC4(k+1),Ibal1(k+1),Ibal2(k+1),Ibal3(k+1),Ibal4(k+1),Ich(k+1));
    k=k+1;
    t(k)=t(k-1)+dt;
end
il1(k)=0;
```

```
il2(k)=0;
il3(k)=0;
il4(k)=0;
iT(k)=0;
```

تخلیه سلول شماره ۱ که دارای بالاتری SOC است تا جایی ادامه می‌یابد که SOC آن به اندازه یک مقدار مشخص (در اینجا 0.005 انتخاب شده است) از SOC سلول شماره ۲ (که SOC آن کمتر از سلول ۱ و بیشتر از بقیه است) پایین تر رود. در واقع یک باند هیستریزیس در اینجا لحاظ کرده‌ایم تا فرکانس سوئیچینگ کاهش یابد. برای پیاده‌سازی هیستریزیس، کد مربوط به دو مرحله فوق در داخل حلقه while زیر قرار می‌گیرد.

```
while (SOC_matrix(B(4))-SOC_matrix(B(3)))> -Mhyst

    . . .
    SOC_matrix=[SOCh1(k+1),SOCh2(k+1),SOCh3(k+1),SOCh4(k+1)];
end
```

به همین ترتیب اگر سلول‌های ۲ و ۳ و ۴ دارای بالاترین SOC باشند نیز مشابه فوق عمل می‌شود و داریم:

```
elseif SOC_matrix(B(4))==SOCh2(k+1)
    while (SOC_matrix(B(4))-SOC_matrix(B(3)))> -Mhyst
        while il2(k)<Imax1
            iT(k+1)=0;
            il2(k+1)=il2(k)+V2(k)*dt/L1;
            il1(k+1)=0;
            il3(k+1)=0;
            il4(k+1)=0;
            Ibal2(k+1)=-il2(k+1)+iT(k+1);
            Ibal1(k+1)=0;
            Ibal3(k+1)=0;
            Ibal4(k+1)=0;

            [V1(k+1),V2(k+1),V3(k+1),V4(k+1),I1(k+1),I2(k+1),I3(k+1),I4(k+1),Ich(k+2),
            SOC1(k+2),SOC2(k+2),SOC3(k+2),SOC4(k+2),Vh1(k+2),Vh2(k+2),Vh3(k+2),Vh4(k+2),
            SOCh1(k+2),SOCh2(k+2),SOCh3(k+2),SOCh4(k+2)] =
            states_Ich_update_SOcbalancing_SMO
            (Vh1(k+1),Vh2(k+1),Vh3(k+1),Vh4(k+1),SOCh1(k+1),SOCh2(k+1),SOCh3(k+1),SOCh
            4(k+1),SOC1(k+1),SOC2(k+1),SOC3(k+1),SOC4(k+1),Ibal1(k+1),Ibal2(k+1),Ibal3
            (k+1),Ibal4(k+1),Ich(k+1));
            k=k+1;
            t(k)=t(k-1)+dt;
        end
        while il2(k)>0
            il2(k+1)=il2(k)-.25*dt*(V1(k)+V2(k)+V3(k)+V4(k))/L1;
            il2(k+1)=il2(k+1)*.5*(1+sign(il2(k+1)));
            il1(k+1)=0;
            il3(k+1)=0;
            il4(k+1)=0;
            iT(k+1)=.25*il2(k+1);
            Ibal1(k+1)=iT(k+1);
            Ibal2(k+1)=iT(k+1);
            Ibal3(k+1)=iT(k+1);
```

```

Ibal4(k+1)=iT(k+1);

[V1(k+1),V2(k+1),V3(k+1),V4(k+1),I1(k+1),I2(k+1),I3(k+1),I4(k+1),Ich(k+2),
SOC1(k+2),SOC2(k+2),SOC3(k+2),SOC4(k+2),Vh1(k+2),Vh2(k+2),Vh3(k+2),Vh4(k+2),
SOCh1(k+2),SOCh2(k+2),SOCh3(k+2),SOCh4(k+2)] =
states_Ich_update_SOCbalancing_SMO
(Vh1(k+1),Vh2(k+1),Vh3(k+1),Vh4(k+1),SOCh1(k+1),SOCh2(k+1),SOCh3(k+1),SOCh
4(k+1),SOC1(k+1),SOC2(k+1),SOC3(k+1),SOC4(k+1),Ibal1(k+1),Ibal2(k+1),Ibal3
(k+1),Ibal4(k+1),Ich(k+1));
k=k+1;
t(k)=t(k-1)+dt;
end
iL1(k)=0;
iL2(k)=0;
iL3(k)=0;
iL4(k)=0;
iT(k)=0;
SOC_matrix=[SOCh1(k+1),SOCh2(k+1),SOCh3(k+1),SOCh4(k+1)];
end
elseif SOC_matrix(B(4))==SOCh3(k+1)
while (SOC_matrix(B(4))-SOC_matrix(B(3)))> -Mhyst
while iL3(k)<Imax1
iT(k+1)=0;
iL3(k+1)=iL3(k)+V3(k)*dt/L1;
iL1(k+1)=0;
iL2(k+1)=0;
iL4(k+1)=0;
Ibal3(k+1)=-iL3(k+1)+iT(k+1);
Ibal1(k+1)=0;
Ibal2(k+1)=0;
Ibal4(k+1)=0;

[V1(k+1),V2(k+1),V3(k+1),V4(k+1),I1(k+1),I2(k+1),I3(k+1),I4(k+1),Ich(k+2),
SOC1(k+2),SOC2(k+2),SOC3(k+2),SOC4(k+2),Vh1(k+2),Vh2(k+2),Vh3(k+2),Vh4(k+2),
SOCh1(k+2),SOCh2(k+2),SOCh3(k+2),SOCh4(k+2)] =
states_Ich_update_SOCbalancing_SMO
(Vh1(k+1),Vh2(k+1),Vh3(k+1),Vh4(k+1),SOCh1(k+1),SOCh2(k+1),SOCh3(k+1),SOCh
4(k+1),SOC1(k+1),SOC2(k+1),SOC3(k+1),SOC4(k+1),Ibal1(k+1),Ibal2(k+1),Ibal3
(k+1),Ibal4(k+1),Ich(k+1));
k=k+1;
t(k)=t(k-1)+dt;
end
while iL3(k)>0
iL3(k+1)=iL3(k)-.25*dt*(V1(k)+V2(k)+V3(k)+V4(k))/L1;
iL3(k+1)=iL3(k+1)*.5*(1+sign(iL3(k+1)));
iL1(k+1)=0;
iL2(k+1)=0;
iL4(k+1)=0;
iT(k+1)=.25*iL3(k+1);
Ibal1(k+1)=iT(k+1);
Ibal2(k+1)=iT(k+1);
Ibal3(k+1)=iT(k+1);
Ibal4(k+1)=iT(k+1);

```

```

[V1(k+1),V2(k+1),V3(k+1),V4(k+1),I1(k+1),I2(k+1),I3(k+1),I4(k+1),Ich(k+2),
SOC1(k+2),SOC2(k+2),SOC3(k+2),SOC4(k+2),Vh1(k+2),Vh2(k+2),Vh3(k+2),Vh4(k+2
),SOCh1(k+2),SOCh2(k+2),SOCh3(k+2),SOCh4(k+2)] =
states_Ich_update_SOCbalancing_SMO
(Vh1(k+1),Vh2(k+1),Vh3(k+1),Vh4(k+1),SOCh1(k+1),SOCh2(k+1),SOCh3(k+1),SOCh
4(k+1),SOC1(k+1),SOC2(k+1),SOC3(k+1),SOC4(k+1),Ibal1(k+1),Ibal2(k+1),Ibal3
(k+1),Ibal4(k+1),Ich(k+1));
k=k+1;
t(k)=t(k-1)+dt;
end
il1(k)=0;
il2(k)=0;
il3(k)=0;
il4(k)=0;
iT(k)=0;
SOC_matrix=[SOCh1(k+1),SOCh2(k+1),SOCh3(k+1),SOCh4(k+1)];
end
elseif SOC_matrix(B(4))==SOCh4(k+1)
while (SOC_matrix(B(4))-SOC_matrix(B(3)))> -Mhyst
while il4(k)<Imax1
iT(k+1)=0;
il4(k+1)=il4(k)+V4(k)*dt/L1;
Ibal4(k+1)=-il4(k+1)+iT(k+1);
il1(k+1)=0;
il2(k+1)=0;
il3(k+1)=0;
Ibal1(k+1)=0;
Ibal2(k+1)=0;
Ibal3(k+1)=0;

[V1(k+1),V2(k+1),V3(k+1),V4(k+1),I1(k+1),I2(k+1),I3(k+1),I4(k+1),Ich(k+2),
SOC1(k+2),SOC2(k+2),SOC3(k+2),SOC4(k+2),Vh1(k+2),Vh2(k+2),Vh3(k+2),Vh4(k+2
),SOCh1(k+2),SOCh2(k+2),SOCh3(k+2),SOCh4(k+2)] =
states_Ich_update_SOCbalancing_SMO
(Vh1(k+1),Vh2(k+1),Vh3(k+1),Vh4(k+1),SOCh1(k+1),SOCh2(k+1),SOCh3(k+1),SOCh
4(k+1),SOC1(k+1),SOC2(k+1),SOC3(k+1),SOC4(k+1),Ibal1(k+1),Ibal2(k+1),Ibal3
(k+1),Ibal4(k+1),Ich(k+1));
k=k+1;
t(k)=t(k-1)+dt;
end
while il4(k)>0
il4(k+1)=il4(k)-.25*dt*(V1(k)+V2(k)+V3(k)+V4(k))/L1;
il4(k+1)=il4(k+1)*.5*(1+sign(il4(k+1)));
iT(k+1)=.25*il4(k+1);
Ibal1(k+1)=iT(k+1);
Ibal2(k+1)=iT(k+1);
Ibal3(k+1)=iT(k+1);
Ibal4(k+1)=iT(k+1);
il1(k+1)=0;
il2(k+1)=0;
il3(k+1)=0;

```

```

[V1(k+1),V2(k+1),V3(k+1),V4(k+1),I1(k+1),I2(k+1),I3(k+1),I4(k+1),Ich(k+2),SOC1
(k+2),SOC2(k+2),SOC3(k+2),SOC4(k+2),Vh1(k+2),Vh2(k+2),Vh3(k+2),Vh4(k+2),SOCh1(
k+2),SOCh2(k+2),SOCh3(k+2),SOCh4(k+2)] =
states_Ich_update_SOCbalancing_SMO
(Vh1(k+1),Vh2(k+1),Vh3(k+1),Vh4(k+1),SOCh1(k+1),SOCh2(k+1),SOCh3(k+1),SOCh4(k+
1),SOC1(k+1),SOC2(k+1),SOC3(k+1),SOC4(k+1),Iba11(k+1),Iba12(k+1),Iba13(k+1),Ib
a14(k+1),Ich(k+1));
k=k+1;
t(k)=t(k-1)+dt;
end
iL1(k)=0;
iL2(k)=0;
iL3(k)=0;
iL4(k)=0;
iT(k)=0;
SOC_matrix=[SOCh1(k+1),SOCh2(k+1),SOCh3(k+1),SOCh4(k+1)];
end
end

```

ب- اختلاف کوچکتر از ۱٪ بوده و نیاز به انجام بالانس نباشد:

در این حالت، جریان بالانس همه سلول ها صفر بوده و هیچ سوئیچی وصل نیست. فقط جریان و ولتاژ ترمینال هر سلول اندازه گیری می شود. جریان گذرنده از همه سلول ها برابر جریان شارژ می باشد.

```

else
Iba11(k+1)=0;
Iba12(k+1)=0;
Iba13(k+1)=0;
Iba14(k+1)=0;
I1(k+1)=Ich(k+1);
I2(k+1)=Ich(k+1);
I3(k+1)=Ich(k+1);
I4(k+1)=Ich(k+1);
V1(k+1)=Voc1+V11+V12+R0*I1(k+1);
V2(k+1)=Voc2+V21+V22+R0*I2(k+1);
V3(k+1)=Voc3+V31+V32+R0*I3(k+1);
V4(k+1)=Voc4+V41+V42+R0*I4(k+1);
iL1(k+1)=0;
iL2(k+1)=0;
iL3(k+1)=0;
iL4(k+1)=0;
k=k+1;
t(k)=t(k-1)+dt;
end

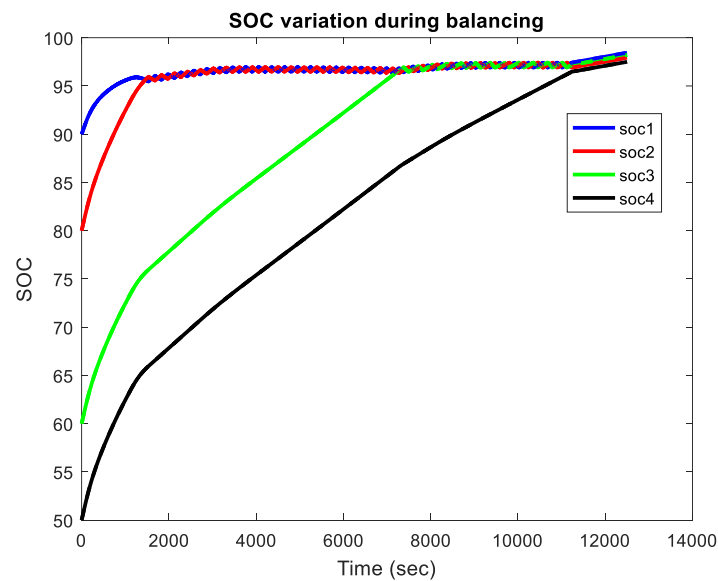
```

این کار آنقدر ادامه می یابد تا پیک باتری کاملاً شارژ شود که شرط شارژ کامل، رسیدن جریان شارژ به یک حد کمینه (در اینجا 1mA) است که در داخل تابع آورده شده است.

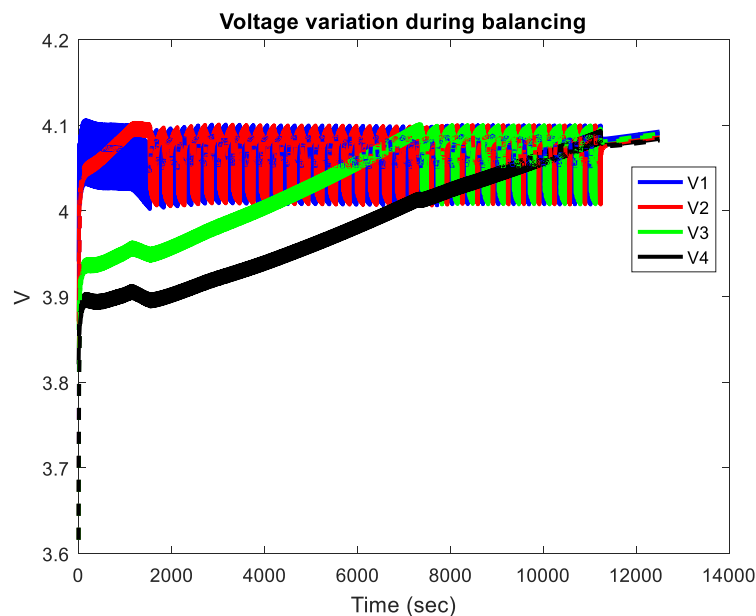
بعد از اتمام حلقه while، به ولتاژ ترمینال تخمین زده شده، مقدار ثابتی (عرض از مبدأ) که از خطی سازی منحنی $V_{oc}(SOC)$ به دست می آید اضافه می شود. به منظور خطی کردن معادلات رویتگر، این مقدار نادیده گرفته شده بود.

$$Vh1 = Vh1+y0; Vh2 = Vh2+y0; Vh3 = Vh3+y0; Vh4 = Vh4+y0;$$

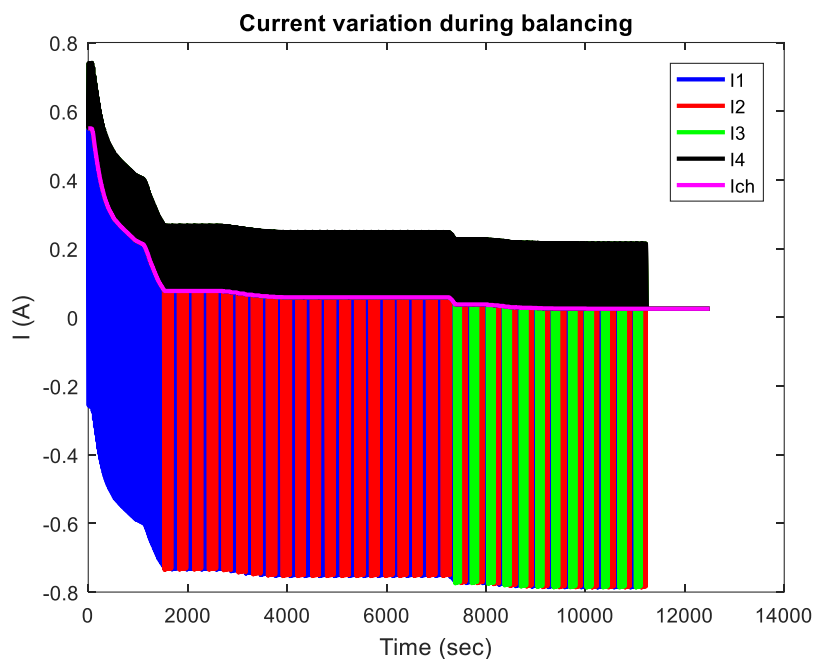
ادامه این کد، دقیقاً مشابه کد مربوط به یکسان‌سازی غیرفعال می‌باشد که در فصل سوم توضیح داده شده است. نتایج اجرای این کد در شکل‌های زیر آمده است. تغییرات زیادی که در ولتاژ و جریان دیده می‌شود ناشی از سوئیچینگ زیاد نیست، بلکه در اثر شارژ و تخلیه انرژی در ترانس است که بستگی به انتخاب اندازه سلف و جریان بیشینه I_{max1} دارد. برای درک بهتر از میزان سوئیچینگ، در شکل ۱۶-۲ تغییرات میانگین ولتاژ سلول‌ها در بازه‌های سوئیچینگ آمده است.



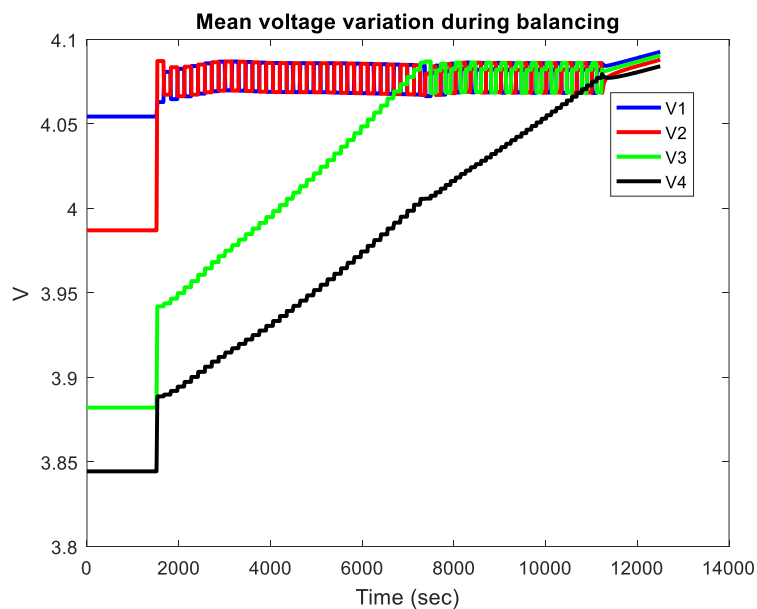
شکل ۱۳-۲ SOC سلول‌ها در طی یکسان‌سازی فعال با روش SMO



شکل ۱۴-۲ ولتاژ سلول‌ها در طی یکسان‌سازی فعال با روش SMO



شکل ۱۵-۲- جریان سلول‌ها در طی یکسان‌سازی فعال با روش SMO



شکل ۱۶-۲- میانگین ولتاژ سلول‌ها در بازه‌های سوئیچینگ در طی یکسان‌سازی فعال با روش SMO

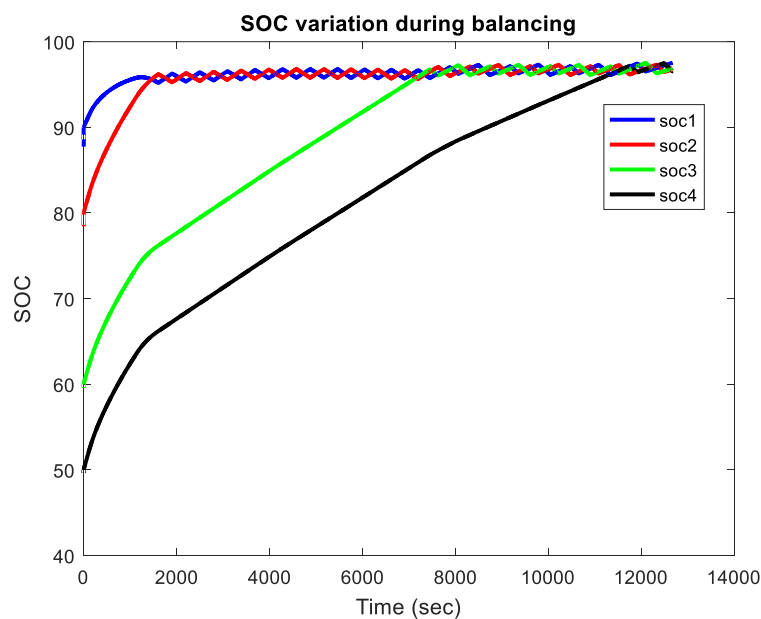
۲-۴-۲- شبیه‌سازی یکسان‌سازی فعال مبتنی بر تخمین سطح شارژ با استفاده از EKF

در این بخش، کد مربوط به شبیه‌سازی یکسان‌سازی شارژ فعال که در آن از تخمینگر EKF برای تخمین SOC استفاده شده است تشریح خواهد شد. در این حالت کدی مشابه کد مربوط به حالت SMO نوشته شده است و فقط چند تفاوت وجود دارد که در فصل سوم به این تفاوت‌ها اشاره شده است. تفاوت اصلی این است که

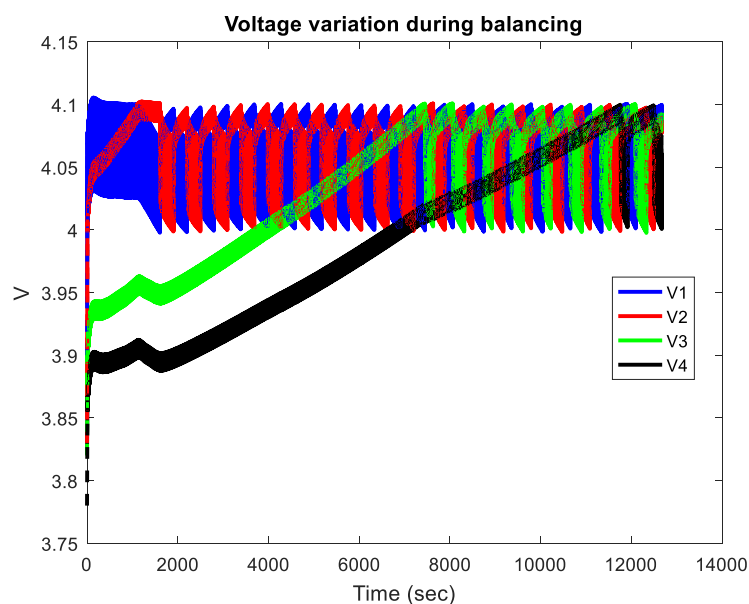
به جای تابع مربوط به SMO، تابع `states_Ich_update_SOCbalancing_EKF` در هر بار فراخوانی می‌شود. جزئیات این تابع در فصل دوم آمده است.

```
function[V1,V2,V3,V4,I1,I2,I3,I4,Ich_new,SOC1_new,SOC2_new,SOC3_new,SOC4_new,Vh1,Vh2,Vh3,Vh4,S0Ch1_new,S0Ch2_new,S0Ch3_new,S0Ch4_new] = ...
states_Ich_update_SOCbalancing_EKF(S0Ch1,S0Ch2,S0Ch3,S0Ch4,SOC1,SOC2,SOC3,SOC4,
Ibal1,Ibal2,Ibal3,Ibal4,Ich)
```

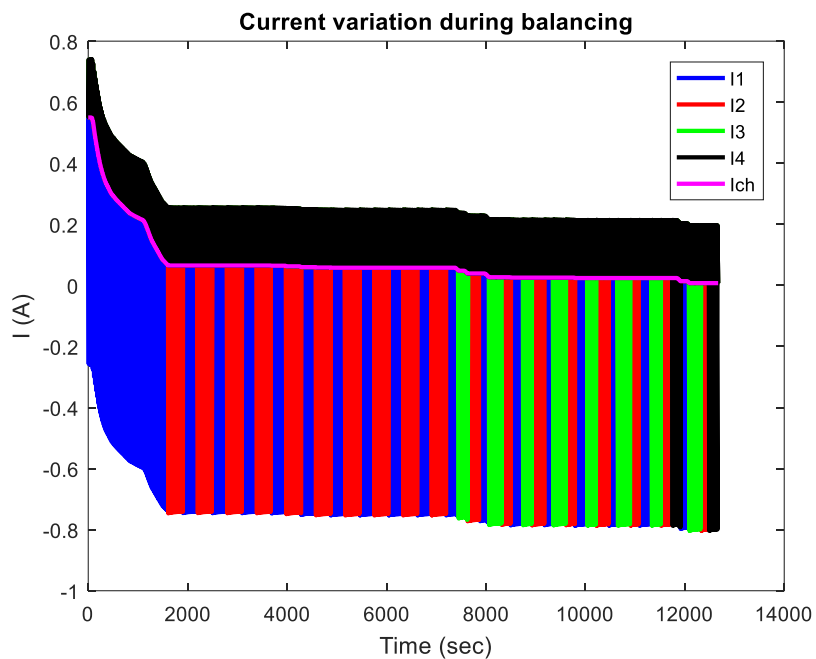
نتایج اجرای این کد در شکل‌های زیر آمده است.



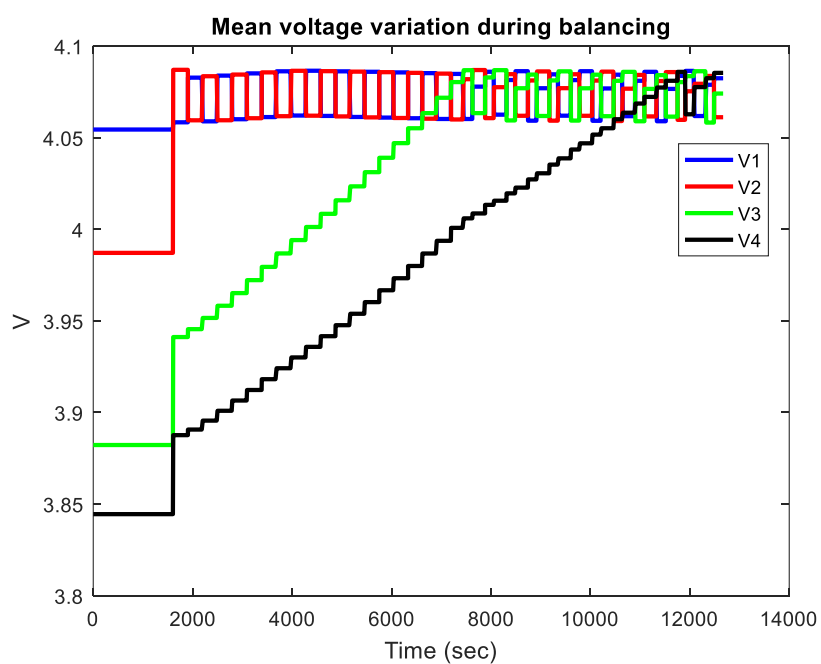
شکل ۱۷-۲- SOC سلول‌ها در طی یکسان‌سازی فعال با روش EKF



شکل ۱۸-۲- ولتاژ سلول‌ها در طی یکسان‌سازی فعال با روش EKF



شکل ۱۹-۲- جریان سلول‌ها در طی یکسان‌سازی فعال با روش EKF



شکل ۲۰-۲- میانگین ولتاژ سلول‌ها در بازه‌های سوئیچینگ در طی یکسان‌سازی فعال با روش EKF

۵-۲- جمع بندی

در این فصل به تشریح کدهای مربوط به شبیه سازی روش های یکسان سازی شارژ غیرفعال و یکسان سازی شارژ فعال مبتنی بر مبدل فلای بک چند-سیم پیچ (که بر روی چهار سلول سری اعمال شده است) در محیط Mfile نرم افزار MATLAB پرداخته شده است. از آنجایی که SOC به طور مستقیم قابل اندازه گیری نیست باید تخمین زده شود. برای تخمین SOC از دو روش رؤیتگر مد لغزشی (SMO) و فیلتر کالمن توسعه یافته (EKF) استفاده شده است. در واقع، هر یک از روش های یکسان سازی شارژ غیرفعال و فعال، یک بار با استفاده از الگوریتم SMO و بار دیگر با استفاده از EKF پیاده سازی شده است.

در ابتدا معادلات فضای حالت مدل الکتریکی باتری که در الگوریتم EKF مورد استفاده قرار می گیرد استخراج شده و سپس الگوریتم EKF و نحوه شبیه سازی آن در MATLAB بیان شده است. در ادامه به استخراج مدل به کار رفته در الگوریتم SMO و نیز معرفی الگوریتم SMO برای تخمین SOC باتری و همچنین نحوه شبیه سازی این روش ها در MATLAB اختصاص یافت.

سپس به تشریح کد مربوط به شبیه سازی روش یکسان سازی شارژ غیرفعال در نرم افزار MATLAB پرداخته شده است. در روش های غیرفعال یکسان سازی شارژ، به هر سلول یک مقاومت اختصاص داده می شود که به صورت موازی با آن قرار می گیرد. مقدار SOC کلیه سلول های استک در هر لحظه با الگوریتم SMO یا EKF تخمین زده شده و به سیستم کنترل اعلام می شود. سیستم کنترل، فرمان وصل را به کلید متصل به سلول با بیشترین SOC ارسال می نماید. شارژ اضافی سلول با SOC بالاتر تلف می شود تا SOC همه سلول های استک به یک سطح برسد.

در ادامه کد مربوط به شبیه سازی روش یکسان سازی شارژ فعال مبتنی بر مبدل های فلای بک چندسیم پیچ در MATLAB تشریح گردیده است. در این یکسان ساز، انرژی اضافی سلول ها به جای اینکه تلف شود، به وسیله مبدل DC به DC ایزوله به سلول های با شارژ کمتر منتقل می شود. نخست مقدار SOC کلیه سلول های استک در هر لحظه با الگوریتم SMO یا EKF تخمین زده شده و به سیستم کنترل اعلام می شود. سیستم کنترل، تمام سلول ها را بررسی کرده و سلول با بیشترین SOC را انتخاب می نماید. با وصل کردن سوئیچ مربوط به آن سلول، انرژی آن در داخل اندوکتانس مغناطیس کنندگی ترانس ذخیره شده و سپس به بقیه سلول ها منتقل می گردد.

۶-۲- مراجع

- [1] M. Chen, and G. A. Rincon-Mora, "Accurate electrical battery model capable of predicting runtime and I-V performance," *IEEE Trans. Energy Convers.*, vol. 21, no. 2, pp. 504-511, Jun. 2006.
- [2] N. A. Chaturvedi, R. Klein, J. Christensen, J. Ahmed, and A. Kojic, "Algorithms for Advanced Battery-Management Systems: Modeling, estimation, and control challenges for lithium-ion battery," *IEEE Control Syst. Mag.*, vol. 30, no. 3, pp. 49-68, Jun. 2010.
- [3] A. R. Pishrobat, "Estimating the parameters of charge, discharge and state of health of the battery using identification and nonlinear estimation techniques, MSc. Thesis, University of Tehran, electrical and Computer School of Engineering, Sep. 2007.
- [4] R. J. Grube, Automotive Battery State-of-Health Monitoring Methods, M.Sc. Thesis, Wright State University, Dec. 2008.
- [5] B. T. Kuhn, G. E. Pitel, and P. T. Krein, "Electrical properties and equalization of lithium-ion cells in automotive applications," in *Vehicle Power and Propulsion IEEE Conference*, 2005.
- [6] J. Zhao, J. Jiang, and L. Niu, "A novel charge equalization technique for electric vehicle battery system," *International Conference on Power Electronics and Drive Systems*, vol. 2, pp. 853-857, 2003.
- [7] M. Gholizadeh, F.R. Salmasi, "Estimation of state of charge, unknown nonlinearities, and state of health of a lithium-ion battery based on a comprehensive unobservable model," *IEEE Transaction on Industrial Electronics*, Vol. 61, No. 3, march 2014.
- [8] J. W. Shin, G. S. Seo, C.Y. Chun, and B. H. Cho, "Selective flyback balancing circuit with improved balancing speed for series connected Lithium-ion batteries," in *Power Electronics Conference (IPEC)*, pp. 1180-1184, 2010.
- [9] A. Imtiaz and F. H. Khan, "Time Shared Flyback Converter Based Regenerative Cell Balancing Technique for Series Connected Li-Ion Battery Strings," *IEEE Trans. Power Electron.*, vol. 28, no. 12, pp. 5960-5975, Dec. 2013.
- [10] M. Charkhgard, and M. Farrokhi, "State-of-Charge Estimation for Lithium-Ion Batteries Using Neural Networks and EKF," *IEEE Trans. Ind. Electron.*, vol. 57, no. 12, pp. 4178-4187, Dec. 2010.

۳- طراحی سخت افزاری و پیاده سازی الگوریتم های یکسان سازی شارژ فعال و غیرفعال

۱-۳- مقدمه

در این فصل به تشریح بستر سخت افزاری و نحوه پیاده سازی عملی روش های یکسان سازی شارژ فعال و غیرفعال بر مبنای هر دو روش تخمین SMO و EKF بر روی پردازشگر DSP و همچنین پیاده سازی الگوریتم های شناسایی پارامترهای باتری بر روی میکروکنترلر AVR می پردازد. به این منظور، ابتدا به معرفی مدار شناسایی پارامترهای باتری اختصاص پرداخته و در ادامه، ضمن معرفی اجزاء سخت افزاری مدار یکسان ساز، برنامه نوشته شده بر روی پردازشگر DSP جهت اجرای روش های یکسان سازی شارژ غیرفعال و فعال تشریح می گردد.

۲-۳- طراحی سخت افزاری مدار تست شناسایی پارامترهای مدل باتری و پیاده سازی الگوریتم های شناسایی

الگوریتم های پیشرفته مورد استفاده در یکسان سازی شارژ سلول ها به منظور تخمین دقیق وضعیت شارژ سلول ها نیازمند به اطلاعات پارامترهای مدل باتری هستند. این پارامترها بعضا توسط سازنده باتری ارائه می گردند. اما به دو دلیل وجود مدار شناسایی در سیستم های مدیریت شارژ باتری لازم می باشد: ۱- ممکن است

پارامترهای ارائه شده توسط سازنده بخوبی قابل انطباق با مدل در نظر گرفته شده در طراحی الگوریتم های تخمین سطح شارژ نباشد. ۲- بعضی از پارامترهای باتری با تغییرات دمایی و رطوبت محیط و همچنین با عملکرد باتری در طول عمر آن بطور غیرخطی تغییر میکنند که این نیازمند به به روزرسانی پیوسته پارامترها می باشد.

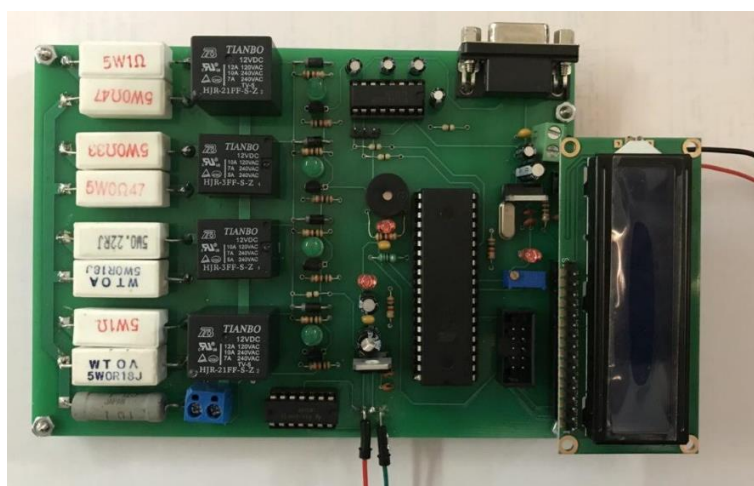
در این فصل مدار طراحی شده برای استخراج داده های مورد استفاده در شناسایی پارامترهای باتری، معرفی شده و کد مورد نظر برای شناسایی پارامترها تشریح می گردد.

۱-۲-۳- اجزای تشکیل دهنده مدار استخراج داده تست جهت شناسایی پارامترهای باتری

ساختار شماتیک مدار تست که بر روی PCB پیاده سازی شده است در شکل ۱-۳ نشان داده شده است. اجزای تشکیل دهنده این مدار شامل: مدار رله و بار قابل برنامه ریزی، پردازشگر، سنسور جریان و ولتاژ، واسط ارتباط سریال، صفحه نمایش و در نهایت باتری مورد استفاده در تست است. در ادامه به تشریح هر یک از این اجزا پرداخته خواهد شد.

مدار رله و بار قابل برنامه ریزی

این مدار تست از یک بار مقاومتی قابل برنامه ریزی با ۴ رله، ۴ دیود نوری، ۴ ترانزیستور و ۴ جفت مقاومت سری تشکیل شده است، به طوری که مقاومت های سری متصل به هر رله به طور موازی با مقاومت های سری متصل به رله های دیگر قرار گرفته است. از ترانزیستورها برای درایو کردن رله ها و از دیودهای نوری برای اطمینان از کارکرد آنها استفاده می شود. با روشن شدن هر رله، دیود نوری مربوط به آن روشن شده و مقاومت های سری متصل به آن وارد مدار دشارژ باتری می شود. هر چه تعداد رله های روشن بیشتر باشد بار مقاومتی دو سر باتری کمتر بوده و جریان آن بیشتر خواهد بود. لذا با کنترل روشن یا خاموش شدن رله ها و مدت زمان روشن بودن آنها می توان جریان دشارژ باتری را کنترل کرده و هر پروفایل دلخواهی از جریان را از باتری کشید. مدار رله و بار قابل برنامه ریزی در بخش سمت چپ مدار تست شکل ۱-۳ قابل مشاهده است.



شکل ۱-۳- مدار تست جهت شناسایی پارامترها

پردازش گر

یک میکروکنترلر AVR سری ATmega16 به عنوان پردازشگر برای کنترل بار مقاومتی و اندازه‌گیری سیگنال‌های جریان و ولتاژ مورد استفاده قرار گرفته است. ترانزیستورهای درایو رله‌ها به ۴ پین پورت D پردازش گر وصل شده و با یک کردن هر پین، رله مربوطه روشن می‌شود. به منظور اندازه‌گیری سیگنال‌های ولتاژ و جریان، دو کانال مبدل آنالوگ به دیجیتال^۱ (ADC) از میکروکنترلر با خروجی دیجیتال ۱۰ بیتی (دقت اندازه‌گیری 0.005 Volt) در نظر گرفته شده است. نرخ نمونه‌برداری از این سیگنال‌ها نیز 20 Hz می‌باشد.

سنسورها

یک مقاومت $1\ \Omega$ که افت ولتاژ دو سر آن بیانگر جریان باتری است با بار مقاومتی سری شده است و لذا از آن می‌توان به عنوان سنسور جریان استفاده کرد. برای اندازه‌گیری جریان، یک سر این مقاومت به یک کانال ADC پردازش گر وصل می‌شود. برای اندازه‌گیری ولتاژ، سر مثبت باتری به یکی دیگر از کانال‌های ADC پردازش گر وصل شده است.

نمایش گر

یک LCD برای نمایش سیگنال‌های جریان و ولتاژ اندازه‌گیری شده لحاظ شده است.

واسط MAX-232

از این واسط برای ایجاد ارتباط بین پردازنده و کامپیوتر و ارسال داده‌های اندازه‌گیری شده به کامپیوتر استفاده شده است. لازم به ذکر است که ارسال داده به کامپیوتر در نرخ ۹۶۰۰ بیت بر ثانیه (bps) صورت می‌گیرد. از آنجایی که این داده‌ها به صورت سریال برای کامپیوتر فرستاده می‌شود، برای دریافت آن‌ها از مبدل سریال به USB استفاده شده است.

باتری

باتری مورد استفاده در این تست، باتری لیتیم-یون با ظرفیت نامی 2000 mAh، ولتاژ نامی 3.7 V، ولتاژ ماکزیمم 4.1 V و ولتاژ قطع 3 V می‌باشد. برای به دست آوردن مشخصه دشارژ باتری، یک بار مقاومتی به آن اعمال می‌شود. همان‌طور که اشاره شد با برنامه‌ریزی این بار می‌توان پروفایل دلخواهی از جریان را تولید کرد. این مشخصه باتری به منظور شناسایی مدل باتری مورد استفاده قرار می‌گیرد.

۲-۲-۳- برنامه استخراج داده تست شناسایی

در این بخش کد نوشته شده بر روی پردازنده AVR جهت استخراج داده‌های مورد نیاز برای شناسایی پارامترهای مدل باتری تشریح می‌گردد. در این کد سیکل‌های دشارژ باتری با ارسال فرمان دشارژ با اتصال یک بار مقاومتی به اندازه ۱.۴ اهم به دو سر باتری اجرا می‌گردد. طول مدت هر سیکل ۱۴ دقیقه است که در آن باتری به مدت ۱۲۰ ثانیه تحت بار قرار گرفته و دشارژ می‌شود و پس از آن به مدت ۷۲۰ ثانیه استراحت می‌کند. در طول مدت دشارژ و در انتهای زمان استراحت ولتاژ و جریان مدار باتری اندازه‌گیری شده و در درگاه خروجی سریال قرار می‌گیرد. با اتصال این درگاه به یک کامپیوتر، این داده‌ها همزمان به صورت پیوسته ذخیره شده و در برنامه شناسایی پارامترها مورد استفاده قرار می‌گیرند.

```
#include <mega16a.h>
```

¹ Analog/Digital Converter

```

#include <stdlib.h>
#include <delay.h>
// Alphanumeric LCD functions
#include <alcd.h>
// Standard Input/Output functions
#include <stdio.h>
// Declare your global variables here
int ii=0, jj=0, kk=0, counter=0, myserial=0, myadc=0, eod=0;
int rest=720, dis=120; //rest and discharge interval per second
float adc_datav, adc_datav1, adc_datai, adc_datai1, Vb, Ib;
char vbig, vsmal, vbig1, vsmal1, ibig, ismal, ibig1, ismal1, lcdv[5], lcdi[5];
#define FIRST_ADC_INPUT 0
#define LAST_ADC_INPUT 1
unsigned int adc_data[LAST_ADC_INPUT-FIRST_ADC_INPUT+1];
// Voltage Reference: AREF pin
#define ADC_VREF_TYPE ((0<<REFS1) | (0<<REFS0) | (0<<ADLAR))
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)
#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<DOR)
// USART Receiver buffer
#define RX_BUFFER_SIZE 48
char rx_buffer[RX_BUFFER_SIZE];
#if RX_BUFFER_SIZE <= 256
unsigned char rx_wr_index=0, rx_rd_index=0;
#else
unsigned int rx_wr_index=0, rx_rd_index=0;
#endif
#if RX_BUFFER_SIZE < 256
unsigned char rx_counter=0;
#else
unsigned int rx_counter=0;
#endif
// This flag is set on USART Receiver buffer overflow
bit rx_buffer_overflow;
// USART Receiver interrupt service routine
interrupt [USART_RXC] void usart_rx_isr(void)
{
char status, data;
status=UCSRA;
data=UDR;
if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)
{
rx_buffer[rx_wr_index++]=data;
#if RX_BUFFER_SIZE == 256
// special case for receiver buffer size=256
if (++rx_counter == 0) rx_buffer_overflow=1;
#else
if (rx_wr_index == RX_BUFFER_SIZE) rx_wr_index=0;
if (++rx_counter == RX_BUFFER_SIZE)
{
rx_counter=0;

```

```
        rx_buffer_overflow=1;
    }
#endif
}
}
#ifdef _DEBUG_TERMINAL_IO_
// Get a character from the USART Receiver buffer
#define _ALTERNATE_GETCHAR_
#pragma used+
char getchar(void)
{
    char data;
    while (rx_counter==0);
    data=rx_buffer[rx_rd_index++];
    if RX_BUFFER_SIZE != 256
    if (rx_rd_index == RX_BUFFER_SIZE) rx_rd_index=0;
    #endif
    #asm("cli")
    --rx_counter;
    #asm("sei")
    return data;
}
#pragma used-
#endif

// USART Transmitter buffer
#define TX_BUFFER_SIZE 48
char tx_buffer[TX_BUFFER_SIZE];
#if TX_BUFFER_SIZE <= 256
unsigned char tx_wr_index=0,tx_rd_index=0;
#else
unsigned int tx_wr_index=0,tx_rd_index=0;
#endif

#if TX_BUFFER_SIZE < 256
unsigned char tx_counter=0;
#else
unsigned int tx_counter=0;
#endif

// USART Transmitter interrupt service routine
interrupt [USART_TXC] void usart_tx_isr(void)
{
    if (tx_counter)
    {
        --tx_counter;
        UDR=tx_buffer[tx_rd_index++];
    }
    if TX_BUFFER_SIZE != 256
    if (tx_rd_index == TX_BUFFER_SIZE) tx_rd_index=0;
    #endif
}
}
```



```

#ifndef _DEBUG_TERMINAL_IO_
// Write a character to the USART Transmitter buffer
#define _ALTERNATE_PUTCHAR_
#pragma used+
void putchar(char c)
{
while (tx_counter == TX_BUFFER_SIZE);
#asm("cli")
if (tx_counter || ((UCSRA & DATA_REGISTER_EMPTY)==0))
{
tx_buffer[tx_wr_index++]=c;
#if TX_BUFFER_SIZE != 256
if (tx_wr_index == TX_BUFFER_SIZE) tx_wr_index=0;
#endif
++tx_counter;
}
else
UDR=c;
#asm("sei")
}
#pragma used-
#endif
// ADC interrupt service routine
// with auto input scanning
interrupt [ADC_INT] void adc_isr(void)
{
static unsigned char input_index=0;
// Read the AD conversion result
adc_data[input_index]=ADCW;
// Select next ADC input
ADMUX=(FIRST_ADC_INPUT | ADC_VREF_TYPE)+input_index;
if (++input_index > (LAST_ADC_INPUT-FIRST_ADC_INPUT))
input_index=0;
// Delay needed for the stabilization of the ADC input voltage
delay_us(10);
// Start the AD conversion
ADCSRA|=(1<<ADSC);
PORTA.7=myadc;
}
// Timer1 overflow interrupt service routine
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
// Place your code here
// Reinitialize Timer1 value = 3msec
TCNT1H=0xff83 >> 8;
TCNT1L=0xff83 & 0xff;

// Place your code here
counter++;
if (counter==3)
{
adc_datav=adc_data[0];
adc_datai=adc_data[1];
}
}

```

```

counter=0;
}
vbig=adc_datav/50;
vsmal=adc_datav-vbig*50;
ibig=adc_datai/50;
ismal=adc_datai-ibig*50;
myserial++;
if (myserial==17)    // sent data to serial port every 17*3=51 msec
{
    putchar(255);
    putchar(vbig);
    putchar(vsmal);
    putchar(200);
    putchar(ibig);
    putchar(ismal);

    myserial=0;
    PORTD.2=~PORTD.2;
    myadc=~myadc;
}

ii++;
if (ii==333) // update LCD every 333*3=999 msec ~ 1 sec
{
    ii=0;
    PORTC.3=~PORTC.3;

    Vb=(float)((adc_datav*4.96)/1023.0);
    ftoa(Vb,3,lcdv);
    lcd_gotoxy(0,0);
    lcd_putsf("Vb=");
    lcd_puts(lcdv);

    Ib=(float)((adc_datai*4.96)/1023.0);
    ftoa(Ib,3,lcdi);
    lcd_gotoxy(0,1);
    lcd_putsf("Ib=");
    lcd_puts(lcdi);

    /*sprintf(lcdv,"Vb=%.3f",Vb);
    lcd_gotoxy(0,0);
    lcd_puts(lcdv);
    sprintf(lcdi,"Ib=%.3f",Ib);
    lcd_gotoxy(0,1);
    lcd_puts(lcdi);*/

    jj++;
    kk++;
    if (jj==rest) {PORTD.4=1;} //rest=rand()/180+600; //rand() generates a pseudo-random
    number between 0 and 32767
    if (jj==dis+rest) {PORTD.4=0; jj=0; } //dis=rand()/136+120;
    if(kk>rest & adc_datav<=600) {eod=1; kk=0;} // End of Discharge

```

```

if (eod==1){
PORTD.4=0;
if(kk==rest){
putchar(100);
PORTD.7=1;
delay_ms(5000);

lcd_gotoxy(0,0);
lcd_putsf("Vb=");
lcd_puts(lcdv);
lcd_gotoxy(8,0);
lcd_putsf(",Ib=");
lcd_puts(lcdi);
lcd_gotoxy(0,1);
lcd_putsf("End of Discharge");

PORTD.7=0;
TIMSK=0x00;}
}
}
}

void main(void)
{
// Declare your local variables here
// Input/Output Ports initialization
// Port A initialization
// Function: Bit7=Out Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
DDRA=(1<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) | (0<<DDA2) | (0<<DDA1)
| (0<<DDA0);
// State: Bit7=0 Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) | (0<<PORTA3) |
(0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);
// Port B initialization
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
DDRB=(0<<DDB7) | (0<<DDB6) | (0<<DDB5) | (0<<DDB4) | (0<<DDB3) | (0<<DDB2) | (0<<DDB1)
| (0<<DDB0);
// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) | (0<<PORTB3) |
(0<<PORTB2) | (0<<PORTB1) | (0<<PORTB0);
// Port C initialization
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=Out Bit2=In Bit1=In Bit0=In
DDRC=(0<<DDC7) | (0<<DDC6) | (0<<DDC5) | (0<<DDC4) | (1<<DDC3) | (0<<DDC2) | (0<<DDC1)
| (0<<DDC0);
// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=0 Bit2=T Bit1=T Bit0=T
PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) |
(0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);
// Port D initialization
// Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=In Bit0=In
DDRD=(1<<DDD7) | (1<<DDD6) | (1<<DDD5) | (1<<DDD4) | (1<<DDD3) | (1<<DDD2) | (0<<DDD1)
| (0<<DDD0);
// State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=T Bit0=T
PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (0<<PORTD3) |
(0<<PORTD2) | (0<<PORTD1) | (0<<PORTD0);

```

```

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 125.000 kHz
// Mode: Ph. correct PWM top=0x00FF
// OC1A output: Disconnected
// OC1B output: Disconnected
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer Period: 4.08 ms
// Timer1 Overflow Interrupt: On
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) |
(1<<WGM10);
TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) | (1<<CS11) |
(1<<CS10);
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) | (1<<TOIE1) |
(0<<OCIE0) | (0<<TOIE0);
// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 2400
UCSRA=(0<<RXC) | (0<<TXC) | (0<<UDRE) | (0<<FE) | (0<<DOR) | (0<<UPE) | (0<<U2X) |
(0<<MPCM);
UCSRB=(1<<RXCIE) | (1<<TXCIE) | (0<<UDRIE) | (1<<RXEN) | (1<<TXEN) | (0<<UCSZ2) |
(0<<RXB8) | (0<<TXB8);
UCSRC=(1<<URSEL) | (0<<UMSEL) | (0<<UPM1) | (0<<UPM0) | (0<<USBS) | (1<<UCSZ1) |
(1<<UCSZ0) | (0<<UCPOL);
UBRRH=0x00;
UBRRL=0xCF;
// Analog Comparator initialization
// Analog Comparator: Off
// The Analog Comparator's positive input is
// connected to the AIN0 pin
// The Analog Comparator's negative input is
// connected to the AIN1 pin
ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) | (0<<ACIS1) |
(0<<ACIS0);
// ADC initialization
// ADC Clock frequency: 125.000 kHz
// ADC Voltage Reference: AREF pin

```

```
// ADC Auto Trigger Source: Free Running
ADMUX=FIRST_ADC_INPUT | ADC_VREF_TYPE;
ADCSRA=(1<<ADEN) | (1<<ADSC) | (1<<ADATE) | (0<<ADIF) | (1<<ADIE) | (1<<ADPS2) |
(1<<ADPS1) | (0<<ADPS0);
SFIOR=(0<<ADTS2) | (0<<ADTS1) | (0<<ADTS0);

// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTC Bit 0
// RD - PORTC Bit 1
// EN - PORTC Bit 2
// D4 - PORTC Bit 4
// D5 - PORTC Bit 5
// D6 - PORTC Bit 6
// D7 - PORTC Bit 7
// Characters/line: 16
lcd_init(16);
lcd_clear();

// Global enable interrupts
#asm("sei")
while (1)
{
    // Place your code here
}
}
```

۳-۲-۳- برنامه شناسایی پارامترهای باتری

در این بخش کد نوشته شده در محیط Matlab جهت شناسایی پارامترهای مدل باتری تشریح می‌گردد. در این کد ابتدا داده‌های تست دشارژ باتری فراخوانی شده سپس الگوریتم‌های شناسایی بر روی این داده‌ها اعمال می‌گردد.

```
dd = 1; %input('Select data between [1,2,3,4,5,6,7]: ');
load(['test_data',num2str(dd),'_modified']);

k0=1;
i0=length(VB);
V = VB(k0:i0);    OCV = Voc(k0:i0);    Vf = V_filtered(k0:i0);
I = IB(k0:i0);    soc = Vsoc(k0:i0);    If = I_filtered(k0:i0);
t = tt(k0:i0);

figure,
plot(t,V,'r','Linewidth',2);hold on,grid on; plot(t,I,'b','Linewidth',2);
plot(t,Vf,'k--','Linewidth',2);hold on,grid on; plot(t,If,'k--','Linewidth',2);
xlabel('Time [sec]');ylabel('V & I');legend('V','I','Vf','If');

%% ***** OCV *****
figure
plot(soc,OCV,'Linewidth',2),grid on;
xlabel('SOC'); ylabel('OCV [V]');
% close all
```

```

%% ***** R0 Estimation *****
zindx = []; %find(diff(I)<=Imax/2);
indx0 = setdiff([1:length(I)],zindx);
rise_ind = find(diff(I)>Imax/2); % rising edge of I
fall_ind = find(diff(I)<-Imax/2); % falling edge of I
figure,
subplot(211),hold on,grid on;box on;
plot(t(rise_ind),I(rise_ind),'rs','Linewidth',2);
plot(t(rise_ind+1),I(rise_ind+1),'ko','Linewidth',2);
plot(t,I,'b','Linewidth',2);xlabel('Time [sec]');ylabel('I [A]');
legend('I_{min,i}','I_{max,i}');xlim([0 5500]);
subplot(212),hold on,grid on;box on;
plot(t(rise_ind),V(rise_ind),'rs','Linewidth',2);
plot(t(rise_ind+1),V(rise_ind+1),'ko','Linewidth',2);
plot(t,V,'b','Linewidth',2);xlabel('Time [sec]');ylabel('V [V]');
legend('V_{max,i}','V_{min,i}');xlim([0 5500]);ylim([3.1 4.2]);
for i=1:length(rise_ind)
    R0n(i) = [mean(V(rise_ind(i)-20:rise_ind(i)))-
mean(V(rise_ind(i):rise_ind(i)+20))]./[mean(I(rise_ind(i):rise_ind(i)+20))-
mean(I(rise_ind(i)-20:rise_ind(i)))];
end
R01 = mean(R0n)
R02 = mean([V(rise_ind)-V(rise_ind+1)]./[I(rise_ind+1)-I(rise_ind)]);
R0 = R02

%% ***** RC Networks Estimation *****
sdata = 'nf'; %input('Select data type ([nf] NOT Filtered & [f] Filtered): ','s');
if sdata=='nf'
    V1 = V(indx0);
    I1 = I(indx0);
elseif sdata=='f'
    V1 = Vf(indx0);
    I1 = If(indx0);
end
Voc1 = OCV(indx0);
t1 = t(1:length(V1));
v1 = (-V1+Voc1-R0*I1);
method = 'armax';%input('Select an estimation method ("arx","armax","arx1 for
RC1","armax1 for RC1"): ','s');
switch method
case 'arx'
    %***** ARX Method *****
    na=2; nb=2; nk=1;
    %%%%%%%%% v1[n] = ((b1*q^-1 + b2*q^-2)/(1 + a1*q^-1 + a2*q^-2)) I[n] %%%%%%%%%
    data=iddata(v1,I1,dt);
    model=arx(data,[na nb nk])
    % model.Algorithm , model.EstimationInfo
    [num den]=tfdata(model,'v');
    b1=num(2); b2=num(3); a1=den(2); a2=den(3);
    y1 = sim(model,I1');
    figure,subplot(211),plot(t1,v1,'r'); hold on; plot(t1,y1,'k-.');ylim([-0.2
0.2]);

```

```

        legend('real output','model output');xlabel('Time'),ylabel('Output');
        subplot(212),plot(t,v1'-y1,'b');legend('Estimation Error');
        xlabel('Time');ylim([-0.1 0.1]);
    case 'armax'
        %***** ARMAX Method *****
        na=2; nb=2; nk=1; % nc=1;
        nc = input('Select noise polynomial order [1,2]: ');
        %% (1 + a1*q^-1 + a2*q^-2)v1[k] = (b1*q^-1 + b2*q^-2)I[k] + (1+c1*q^-1 +
        c2*q^-2) n[k]%%%%%%%%
        data=iddata(v1',I1',dt);
        model=armax(data,[na nb nc nk])
        % model.Algorithm , model.EstimationInfo
        [num den]=tfdata(model,'v');
        b1=num(2); b2=num(3); a1=den(2); a2=den(3);
        y1 = sim(model,I1');
        figure,subplot(211),plot(t1,v1,'r'); hold on; plot(t1,y1,'k-.'); ylim([-0.2
        0.2]);
        legend('real output','model output');xlabel('Time'),ylabel('Output');
        subplot(212),plot(t,v1'-y1,'b');legend('Estimation Error');
        xlabel('Time');ylim([-0.1 0.1]);
    case 'arx1'
        %***** ARX Method *****
        na=1; nb=1; nk=1;
        %%%%%%%%% v1[n] = ((b1*q^-1)/(1 + a1*q^-1)) I[n] %%%%%%%%%
        data=iddata(v1',I1',dt);
        model=arx(data,[na nb nk])
        % model.Algorithm , model.EstimationInfo
        [num den]=tfdata(model,'v');
        b1=num(2); a1=den(2);
        y1 = sim(model,I1');
        figure,subplot(211),plot(t1,v1,'r'); hold on; plot(t1,y1,'k-.');ylim([-0.2
        0.2]);
        legend('real output','model output');xlabel('Time'),ylabel('Output');
        subplot(212),plot(t,v1'-y1,'b');legend('Estimation Error');
        xlabel('Time');ylim([-0.1 0.1]);
    case 'armax1'
        %***** ARMAX Method *****
        na=1; nb=1; nk=1; %nc=1;
        nc = input('Select noise polynomial order [1,2]: ');
        %% (1 + a1*q^-1)v1[k] = (b1*q^-1)I[k] + (1+c1*q^-1) n[k]%%%%%%%%
        data=iddata(v1',I1',dt);
        model=armax(data,[na nb nc nk])
        % model.Algorithm , model.EstimationInfo
        [num den]=tfdata(model,'v');
        b1=num(2); a1=den(2);
        y1 = sim(model,I1');
        figure,subplot(211),plot(t1,v1,'r'); hold on; plot(t1,y1,'k-.'); ylim([-0.2
        0.2]);
        legend('real output','model output');xlabel('Time'),ylabel('Output');
        subplot(212),plot(t,v1'-y1,'b');legend('Estimation Error');
        xlabel('Time');ylim([-0.1 0.1]);
end
%*****

```

```
switch method
case {'arx','armax'}
    Tp1 = -dt/log(-a1/2-sqrt((-a1/2)^2-a2));
    Tp2 = -dt/log(-a1/2+sqrt((-a1/2)^2-a2));
    Kp = b1;
    Tz = -b2/b1;
    R1 = Kp*(exp(-dt/Tp1)-Tz)/(exp(-dt/Tp1)-exp(-dt/Tp2))/(1-exp(-dt/Tp1));
    R2 = Kp*(exp(-dt/Tp2)-Tz)/(exp(-dt/Tp2)-exp(-dt/Tp1))/(1-exp(-dt/Tp2));
    C1 = Tp1/R1;
    C2 = Tp2/R2;
    [R1 R2]
    [C1 C2]
    [Tp1 Tp2]
case {'arx1','armax1'}
    Tp1 = -dt/log(-a1);
    R1 = b1/(1-exp(-dt/Tp1));
    C1 = Tp1/R1;
    [R1]
    [C1]
    [Tp1]
End
%% ***** Evaluating Model *****
y1 = sim(model,I1');
V_est = Voc1'-R0*I1'-y1;
b1 = R1*(1-exp(-dt/Tp1))+R2*(1-exp(-dt/Tp2));
b2 = -R1*(1-exp(-dt/Tp1))*exp(-dt/Tp2)-R2*(1-exp(-dt/Tp2))*exp(-dt/Tp1);
a1 = -exp(-dt/Tp1)-exp(-dt/Tp2);
a2 = exp(-dt/Tp1)*exp(-dt/Tp2);
A = [1 a1 a2];
B = [0 b1 b2];
m0 = idpoly(A,B);
y = sim(m0,IB');
V_est = Voc_fit'-R0*IB'-y;
figure,
subplot(211),plot(tt,V1,'b',tt,V_est,'r','Linewidth',2);
hold on,grid on,
ylabel('Voltage [V]');
legend('True','Model');ylim([3.4 4.2]);
subplot(212),plot(tt,V1'-V_est,'r','Linewidth',2);grid
ylabel('Error [V]');
xlabel('Time [sec]');
ylim([-0.05 0.05]);
```

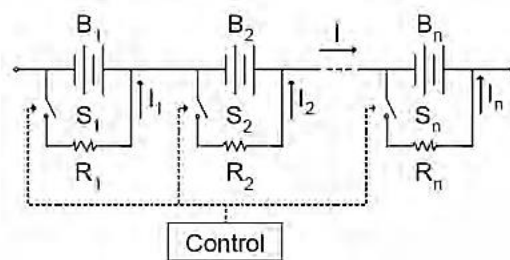
۳-۳- برنامه نویسی و پیاده‌سازی عملی الگوریتم‌های یکسان‌سازی شارژ

در این فصل سخت‌افزار مدار یکسان‌سازی شارژ معرفی می‌شود و کدهای توسعه داده شده جهت پیاده‌سازی بر روی پردازنده DSP تشریح می‌گردند.

۳-۳-۱- یکسان‌سازی غیرفعال

در روش‌های غیر فعال یکسان‌سازی شارژ، به هر سلول یک مقاومت اختصاص داده می‌شود که به صورت موازی با آن قرار می‌گیرد (شکل ۳-۲). در این روش، ولتاژ کلیه سلول‌های استک در هر لحظه به سیستم

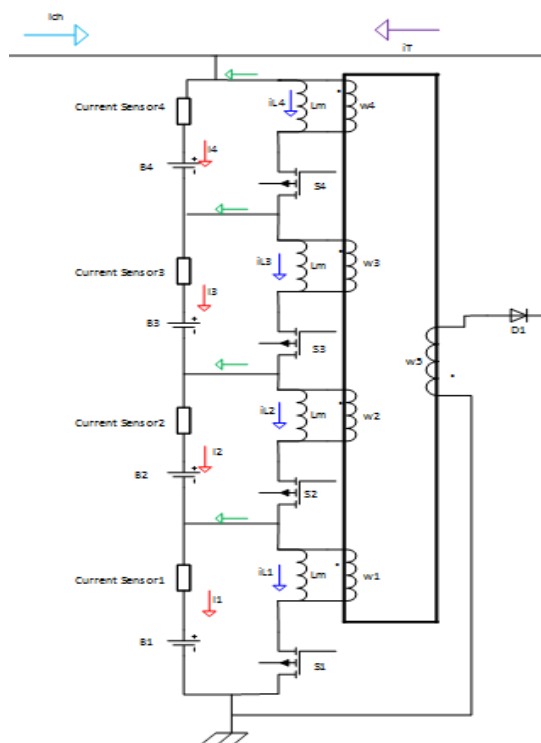
کنترل اعلام شده و سیستم کنترل، فرمان وصل را به کلید متصل به سلول با بیشترین ولتاژ ارسال می‌نماید [۶]. شارژ اضافی سلول با ولتاژ بالاتر تلف می‌شود تا ولتاژ همه سلول‌های استک به یک سطح برسد. مقدار جریانی که به عناصر شنت می‌رسد با ولتاژ سلول متناسب است؛ بنابراین هر چه ولتاژ سلول افزایش یابد مقدار جریانی که به شنت فرستاده می‌شود بیشتر می‌شود. سرعت یکسانسازی با انتخاب مقاومت شانت مناسب قابل تنظیم می‌باشد.



شکل ۲-۳- مدار یکسانساز شارژ غیرفعال

۲-۳-۳- یکسانسازی فعال

در این یکسانسازها، انرژی اضافی سلول‌ها به جای اینکه تلف شود، به وسیله‌ی عناصر فعالی نظیر خازن، مبدل‌های DC به DC ایزوله و یا غیر ایزوله و یا ترکیبی از آن‌ها به سلول‌های با شارژ کمتر منتقل گشته و به این ترتیب باعث بالا رفتن بازدهی این روش‌ها نسبت به روش‌های یکسانسازی غیرفعال خواهد شد. در این پروژه از روش یکسانسازی شارژ فعال با مبدل مبتنی بر ترانسفرمر ایزوله (مبدل فلای‌بک مرکزی) استفاده شده است. عنصر اصلی این مبدل‌های فلای‌بک، یک هسته با دو و یا چند سیم‌پیچ است. شکل ۳-۳ نمایش‌دهنده‌ی مدار یکسانساز مبتنی بر مبدل‌های فلای‌بک چندسیم‌پیچه می‌باشند که در این پروژه مورد استفاده قرار گرفته است [۸]. در گزارش فاز دوم به بررسی حالات مختلف عملکردی این مدار یکسانساز پرداخته شده است.



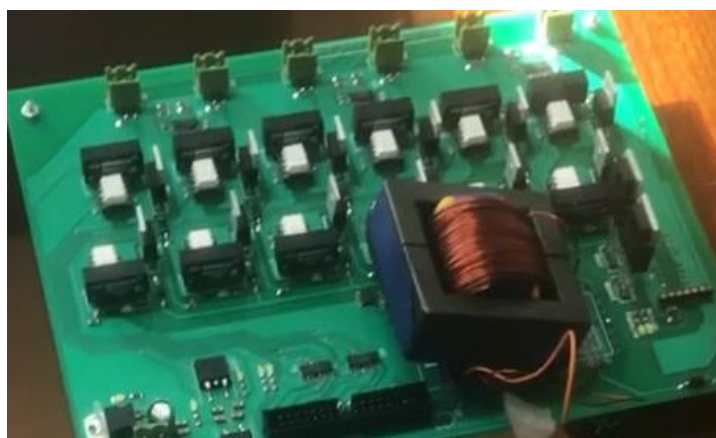
شکل ۳-۳- یکسانساز با مبدل فلای بک مرکزی با سیم‌پیچ‌های ثانویه متعدد

۳-۳-۳- اجزای تشکیل‌دهنده مدار تست جهت یکسان‌سازی شارژ

ساختار شماتیک مدار تست که بر روی PCB پیاده‌سازی شده است در شکل ۳-۴ نشان داده شده است. در ادامه به تشریح اجزای این مدار پرداخته خواهد شد.

مدار سوئیچینگ برای انتخاب سلول باتری

برای انتخاب هر یک از سلول‌های باتری از یک مدار سوئیچینگ بر سر راه آن استفاده شده است. هر کدام از این مدارهای سوئیچینگ شامل دو اپتوکوپلر سری HCPL3120، یک مبدل DC به DC (MinMax) سری



شکل ۳-۴- مدار تست جهت یکسان‌سازی شارژ

MAU200، دو سوئیچ Mosfet سری IFR640 می‌باشد. با تحریک اپتوکوپلر متصل به هر سلول باتری، آن سلول در مدار دشارژ قرار می‌گیرد.

مدار انتخاب نوع یکسان‌سازی (فعال/غیرفعال)

برای انتخاب نوع یکسان‌سازی فعال یا غیرفعال از یک مدار سوئیچینگ شامل دو سوئیچ IGBT سری GW38IH130D، یک IC معکوس‌کننده سیگنال با مدل M74HC04M1R، یک MinMax و دو اپتوکوپلر سری HCPL3120 برای تحریک IGBT مربوطه استفاده شده است.

پردازش‌گر

یک میکروکنترلر DSP سری TMS320F2833x به عنوان پردازشگر برای کنترل شارژ و دشارژ باتری‌ها و اندازه‌گیری سیگنال‌های جریان و ولتاژ مورد استفاده قرار گرفته است. به منظور خواندن سیگنال‌های ولتاژ و جریان، ۸ کانال مبدل آنالوگ به دیجیتال (ADC) از میکروکنترلر با خروجی دیجیتال 12 بیتی در نظر گرفته شده است. نرخ نمونه‌برداری از این سیگنال‌ها نیز 1 Hz می‌باشد، یعنی هر ۱ ثانیه یک بار این سیگنال‌ها به پورت سریال ارسال شده و در داخل کامپیوتر برای تحلیل ذخیره می‌شود.

سنسورها

از دو سنسور جریان سری ACS712 برای اندازه‌گیری جریان‌های اولیه و ثانویه ترانس استفاده شده و خروجی هر سنسور به یکی از کانال‌های ADC متصل گردیده است. برای اندازه‌گیری ولتاژ هر باتری نیز از یک مدار آپامپی با آپامپ‌های سری MCP600 بین ترمینال باتری و کانال ADC برای تضعیف ولتاژ استفاده شده است.

مدار ایزولاسیون

برای ایزولاسیون و جلوگیری از جریان‌کشی و آسیب به پردازشگر DSP از یک مدار ایزولاسیون با IC سری HEF4050B استفاده شده است.

رگولاتور ولتاژ

در این مدار از رگولاتورهای ولتاژ سری 78xx برای ساخت ولتاژهای مورد نیاز (۵، ۱۲ و ۱۵ ولت) از روی ولتاژ ورودی 24V استفاده شده است.

باتری

در این مدار تست از ۶ سلول سری استفاده شده است. باتری مورد استفاده در این تست، باتری لیتیوم-یون با ظرفیت نامی 2000 mAh، ولتاژ نامی 3.7 V، ولتاژ ماکزیمم 4.1 V و ولتاژ قطع 3 V می‌باشد.

۴-۳-۳- پیاده‌سازی روش‌های یکسان‌سازی بر روی DSP

در ادامه، کدهای نوشته شده برای پردازشگر DSP در نرم‌افزار Code Composer مربوط به روش یکسان‌سازی شارژ فعال و غیرفعال شش باتری سری با استفاده از هر دو روش SMO و EKF تشریح خواهد شد.

خطوط زیر برخی از فایل‌های هدر مربوط به کتابخانه‌ها، توابع ریاضی و توابع ورودی-خروجی را به کد اضافه کرده و ساختار توابع استفاده شده در برنامه را بیان می‌کند، به عنوان مثال، تابع init_Gpio پورت‌های ورودی-خروجی (IO) را فعال می‌کند و scia_fifo_init به منظور راه‌اندازی پورت سریال است. تابع scia_msg یک ورودی به صورت رشته دریافت کرده و آن را بر روی پورت سریال قرار می‌دهد. در توابع

ترتیب تخمین‌گرهای EKF و SMO پیاده‌سازی شده است. states_Ich_update_SOCbalancing_EKF و states_Ich_update_SOCbalancing_SMO به

```
#include "DSP28x_Project.h"
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
// Prototype statements for functions found within this file.
interrupt void adc_isr(void);
interrupt void cpu_timer0_isr(void);
void median(int, int[]);
void initLCD(void);
void enable(void);
void LCD_WriteChar(char c);
void LCD_WriteString(char * string);
void LCD_Cmd(unsigned int cmd);
void states_Ich_update_SOCbalancing_EKF(void);
void states_Ich_update_SOCbalancing_SMO(void);
// Prototype statements for functions found within this file.
void scia_echoback_init(void);
void scia_fifo_init(void);
void scia_xmit(int a);
void scia_msg(char *msg);
void init_Gpio(void);
void Display(void);
```

در خطوط زیر از کد، برخی از متغیرها جهانی (Global) تعریف شده‌اند تا در همه جای برنامه امکان دسترسی به مقدار آن وجود داشته باشد. برخی از این متغیرها که در تابع اصلی برنامه (Main) مورد استفاده قرار گرفته‌اند در جدول ۱-۳ توصیف شده‌اند.

```
// Global variables used in this example:
Uint16 Enableflag=1;
Uint16 z=0, LoopCount;
Uint16 ConversionCount;
Uint16 Voltage1[1];
Uint16 Voltage2[1];
Uint16 Voltage3[1];
Uint16 Voltage4[1];
Uint16 Voltage5[1];
Uint16 Voltage6[1];
Uint16 Current1[1];
Uint16 Current2[1];

float Ip_real, Is_real;
Uint16 VB1, VB2, VB3, VB4, VB5, VB6;
Uint16 SOC1, SOC2, SOC3, SOC4, SOC5, SOC6, SOCmax, SOCmin;

Uint16 ADC_Read=0;
Uint16 Vmax, Vmin;
Uint16 PINout;
Uint32 TIMER_COUNT, Temp=0;
Uint16 Max_Bat=1, Min_Bat=1;
int i, j, ADC_count=100;
```

```

Uin16 disp2,disp3,disp5,disp6,disp8,disp9,disp11,disp12,
disp14,disp15,disp17,disp18;
int disp1,disp4,disp7,disp10,disp13,disp16;
char str[300]="hello";

```

جدول ۱-۳- متغیرهای جهانی به کار رفته در تابع اصلی (Main)

متغیر	توصیف متغیر	متغیر	توصیف متغیر
Voltage1...Voltage6	ولتاژهای خوانده شده توسط ADC (mV)	VB1...VB6	ولتاژهای دو سر هر سلول (mV)
Current1	جریان سلول در حال دشارژ (mA)	Current2	جریان ورودی استک باتری (mA)
Is_real, Ip_real	جریان اولیه و ثانویه ترانس	SOC1...SOC6	SOC هر سلول (میلی واحد)
disp1...disp18	متغیرهایی برای ارسال به پورت سریال	SOCmax, SOCmin	SOC ماکزیمم و مینیمم
Max_Bat, Min_Bat	سلول‌های با SOC ماکزیمم و مینیمم	PINout	فرمان On/Off سوئیچ‌ها

برخی دیگر از متغیرهای جهانی نیز هستند که هم در تابع Main و هم در توابع مربوط به تخمین‌گرهای SMO یا EKF مورد استفاده قرار گرفته‌اند که در جدول ۲-۳ گرد آمده است.

```

//*****
float PI=3.1415, dt=0.05;
float Rsd=200, R0=0.1055, R1=0.0021, R2=0.0378, C1=80, C2=1311.1, Cb=7200, a0=3.661,
a1=0.333, a2=-0.3601, a3=0.4933, a4=-1.122, b=39.62;
float I1, I2, I3, I4, I5, I6;
float V1, V2, V3, V4, V5, V6;
float Vh1=0, Vh11=0, Vh12=0, SOCh1=0.4;
float Vh2=0, Vh21=0, Vh22=0, SOCh2=0.36;
float Vh3=0, Vh31=0, Vh32=0, SOCh3=0.35;
float Vh4=0, Vh41=0, Vh42=0, SOCh4=0.32;
float Vh5=0, Vh51=0, Vh52=0, SOCh5=0.34;
float Vh6=0, Vh61=0, Vh62=0, SOCh6=0.37;
float Xh1[4], Xh2[4], Xh3[4], Xh4[4], Xh5[4], Xh6[4], Temp_Xh[4];
float EYE[4][4]={0,0,0,0},{0,1,0,0},{0,0,1,0},{0,0,0,1}};

```

جدول ۲-۳- متغیرهای جهانی به کار رفته در Main و تخمین‌گرها

متغیر	توصیف متغیر	متغیر	توصیف متغیر
dt	زمان نمونه‌برداری	EYE	ماتریس واحد
I1...I6	جریان عبوری از هر سلول (آمپر)	V1...V6	ولتاژ دو سر هر سلول (ولت)

Vh1...Vh6	تخمین ولتاژ دو سر هر سلول (ولت)	Vh11...Vh62	تخمین ولتاژ دو سر شبکه‌های RC
Temp_Xh	متغیر کمکی برای محاسبات	SOCh1...SOCh6	تخمین SOC هر سلول
Xh1...Xh6	بردار تخمین متغیرهای حالت هر سلول	R2 ,R1 ,R0 ,Rsd a0,a1,a2,a3,a4,b Cb ,C2 ,C1	پارامترهای مدل باتری

متغیرهای زیر پارامترهای مربوط به رویکرد SMO می‌باشند که در توابع مربوطه مورد استفاده قرار گرفته‌اند.
این پارامترها در ادامه توصیف خواهند شد.

```
float qq[2]={3.6,0.4542}, a0l, a1l, y0, h=0.2, W=0.1, ey, v_ey, yh1;
float HH[4]={0.000915, -0.95233, -0.95206, 0}, p[3]={-5, -0.0005, -0.02};
float BB[4], CC[4]={0,0,1,0}, Cd[4]={0,0,1,0}, TT[4]={0,0,1,0};
float AA[4][4]={{{1,0,0,0},{0,1,0,0},{0,0,1,0},{0,0,0,1}}, Ad[4][4];
```

جدول ۳-۳- پارامترهای رویکرد SMO

متغیر	توصیف متغیر	متغیر	توصیف متغیر
y0, a0l, a1l, qq	ضرایب خطی‌سازی منحنی Voc	W	یک ماتریس مثبت معین
Ad, AA, BB	ماتریس‌های مدل فضای حالت	HH	ماتریس بهره رویکرد
TT	ماتریس ترم عدم قطعیت	h	بهره سوئیچینگ
ey	خطای تخمین ولتاژ	v_ey	تابع سوئیچینگ

متغیرهای زیر، برخی پارامترهای مربوط به تخمینگر EKF می‌باشند و برخی دیگر (Temp) برای انجام محاسبات ماتریسی تعریف شده‌اند. این پارامترها در ادامه توصیف خواهند شد.

```
float OCV, B[3], A[3][3]={{{1,0,0},{0,1,0},{0,0,1}}, H[3]={1,1,1};
float R, Q[3][3] = {{1,0,0},{0,1,0},{0,0,1}};
float P1[3][3]={{{0,0,0},{0,0,0},{0,0,0}}, P2[3][3]={{{0,0,0},{0,0,0},{0,0,0}},
P3[3][3]={{{0,0,0},{0,0,0},{0,0,0}};
float P4[3][3]={{{0,0,0},{0,0,0},{0,0,0}}, P5[3][3]={{{0,0,0},{0,0,0},{0,0,0}},
P6[3][3]={{{0,0,0},{0,0,0},{0,0,0}};
float P1_Temp1[3][3], P2_Temp1[3][3], P3_Temp1[3][3], P4_Temp1[3][3], P5_Temp1[3][3],
P6_Temp1[3][3];
float P1_Temp2[3][3], P2_Temp2[3][3], P3_Temp2[3][3], P4_Temp2[3][3], P5_Temp2[3][3],
P6_Temp2[3][3];
float K_temp=0, K[3];
```

جدول ۳-۴- پارامترهای تخمینگر EKF

متغیر	توصیف متغیر	متغیر	توصیف متغیر
OCV	مقدار ولتاژ مدار باز فعلی	H	بردار مشتق جزئی خروجی
A, B	ماتریس‌های مدل فضای حالت	R, Q	ماتریس کوواریانس نویز فرایند و اندازه‌گیری
K	بهره فیلتر کالمن	P1...P6	ماتریس کوواریانس خطا برای هر سلول

خطوط زیر که مربوط به تابع اصلی برنامه (Main) است به منظور راه‌اندازی برخی از زیربخش‌های DSP می‌باشد.

```
main()
{
// Step 1. Initialize System Control:
// PLL, WatchDog, enable Peripheral Clocks
// This example function is found in the DSP2833x_SysCtrl.c file.
InitSysCtrl();

EALLOW;
#if (CPU_FRQ_150MHZ) // Default - 150 MHz SYSCLKOUT
#define ADC_MODCLK 0x3 // HSPCLK = SYSCLKOUT/2*ADC_MODCLK2 = 150/(2*3) = 25.0
MHz
#endif
#if (CPU_FRQ_100MHZ)
#define ADC_MODCLK 0x2 // HSPCLK = SYSCLKOUT/2*ADC_MODCLK2 = 100/(2*2) = 25.0
MHz
#endif
EDIS;
EALLOW;
SysCtrlRegs.HISPCP.all = ADC_MODCLK;
EDIS;
DINT;
InitPieCtrl();
// Disable CPU interrupts and clear all CPU interrupt flags:
IER = 0x0000;
IFR = 0x0000;
InitPieVectTable();
EALLOW; // This is needed to write to EALLOW protected register
PieVectTable.TINT0 = &cpu_timer0_isr;
PieVectTable.ADCINT = &adc_isr;
EDIS; // This is needed to disable write to EALLOW protected registers

init_Gpio();
scia_fifo_init(); // Initialize the SCI FIFO
scia_echoback_init(); // Initialize SCI for echoback
for(i=0;i<=1000;i++);

//initLCD(); //LCD Now initialized and ready to Print!
scia_msg("\n\rSTART\n\r");
```

خطوط زیر پارامترها و ماتریس های ثابتی که در رؤیتگر SMO وجود دارند را تعریف می کند. پارامترهای به کار رفته در الگوریتم تخمین SMO همگی در جدول ۳-۳ آورده شده است.

```
//***** SMO Constant Parameters*****  
a0l=qq[0];  
a1l=qq[1];  
y0 = a0l;  
AA[0][0]=-1/(Rsd*Cb);  
AA[1][1]=-1/(R1*C1);  
AA[2][0]=-a1l/(Rsd*Cb)+a1l/(R2*C2);  
AA[2][1]=1/(R2*C2)-1/(R1*C1);  
AA[2][2]=AA[3][3]=-1/(R2*C2);  
BB[0]=1/Cb;  
BB[1]=1/C1;  
BB[2]=R0/(R2*C2)+a1l/(Cb)+1/C1+1/C2;  
BB[3]=1/C2;  
for(i=0;i<=3;i++)  
{  
    TT[i] = 100*BB[i];  
    for(j=0;j<=3;j++)  
    {  
        Ad[i][j]=AA[i][j]*dt+EYE[i][j];  
    }  
}
```

خطوط زیر برای انجام تنظیمات LCD و نوشتن بر روی LCD می باشد.

```
//*****  
for(i=0;i<=1000;i++);  
LCD_Cmd(0x0F);  
LCD_Cmd(0x01); //Screen Clear Command , Cursor at Home  
LCD_Cmd(0x10);  
LCD_Cmd(0xC0); //Come to 2nd Row
```

در خطوط زیر تایمر CPU راه اندازی شده و تنظیمات مربوط به ADC برای خواندن مقادیر ولتاژ و جریان انجام می شود.

```
InitCpuTimers(); // For this example, only initialize the Cpu Timers  
InitAdc(); // For this example, init the ADC  
  
#if (CPU_FRQ_150MHZ)  
    ConfigCpuTimer(&CpuTimer0, 150, 500);  
#endif  
#if (CPU_FRQ_100MHZ)  
    ConfigCpuTimer(&CpuTimer0, 100, 500000);  
#endif  
  
CpuTimer0Regs.TCR.all = 0x4001; // Use write-only instruction to set TSS bit = 0  
  
// Step 5. User specific code, enable interrupts:  
  
// Enable ADCINT in PIE  
PieCtrlRegs.PIEIER1.bit.INTx6 = 1;  
PieCtrlRegs.PIEIER1.bit.INTx7 = 1;
```



```

IER |= M_INT1; // Enable CPU Interrupt 1
EINT;          // Enable Global interrupt INTM
ERTM;          // Enable Global realtime interrupt DBGM

// Configure ADC
AdcRegs.ADCMAXCONV.all = 0x00FF;          // Setup 8 conv's on SEQ1

AdcRegs.ADCCHSELSEQ1.bit.CONV00 = 0x6; // Setup ADCINA6 as 1st SEQ1 conv.
AdcRegs.ADCCHSELSEQ1.bit.CONV01 = 0x7; // Setup ADCINA7 as 2nd SEQ1 conv.
AdcRegs.ADCCHSELSEQ1.bit.CONV02 = 0x0; // Setup ADCINA0 as 3rd SEQ1 conv.
AdcRegs.ADCCHSELSEQ1.bit.CONV03 = 0x1; // Setup ADCINA1 as 4th SEQ1 conv.
AdcRegs.ADCCHSELSEQ2.bit.CONV04 = 0x2; // Setup ADCINA2 as 5th SEQ1 conv.
AdcRegs.ADCCHSELSEQ2.bit.CONV05 = 0x3; // Setup ADCINA3 as 6th SEQ1 conv.
AdcRegs.ADCCHSELSEQ2.bit.CONV06 = 0x4; // Setup ADCINA4 as 7th SEQ1 conv.
AdcRegs.ADCCHSELSEQ2.bit.CONV07 = 0x5; // Setup ADCINA5 as 8th SEQ1 conv.

AdcRegs.ADCTRL2.bit.EPWM_SOCA_SEQ1 = 1; // Enable SOCA from ePWM to start SEQ1
AdcRegs.ADCTRL2.bit.INT_ENA_SEQ1 = 1;   // Enable SEQ1 interrupt (every EOS)
EPwm1Regs.ETSEL.bit.SOCAEN = 1;         // Enable SOC on A group

EPwm1Regs.ETSEL.bit.SOCASEL = 4;         // Select SOC from from CPMA on upcount
EPwm1Regs.ETPS.bit.SOCAPRD = 1;         // Generate pulse on 1st event

EPwm1Regs.CMPA.half.CMPA = 0x0020;      // Set compare A value
EPwm1Regs.TBPRD = 0x0040;               // Set period for ePWM1
EPwm1Regs.TBCTL.bit.CTRMODE = 0;        // count up and start

```

در خطوط زیر، رجیسترهای داده IO پاک می‌شود و برنامه در یک حلقه for بینهایت منتظر اعلام یک وقفه از ADC می‌ماند که این وقفه بیانگر اتمام خواندن و تبدیل سیگنال توسط ADC است.

```

// Wait for ADC interrupt
//*****
TIMER_COUNT=0;
EALLOW;
GpioDataRegs.GPACLEAR.all=0xFFFF;
EDIS;

LoopCount = 0;
ConversionCount = 0;
Enableflag=1;
for(;;)
{
    LoopCount++;
}
}

```

با اعلام یک وقفه توسط ADC، برنامه وارد این وقفه شده، از مقادیر خوانده شده توسط ADC استفاده کرده و یک سری عملیات را انجام می‌دهد. کد زیر بخشی از وقفه ADC را نشان می‌دهد که در آن ولتاژها و جریان-های خوانده شده توسط ADC از داخل رجیسترهای ADC برداشته می‌شود.

```

interrupt void adc_isr(void)
{
    ConversionCount=0;
}

```

```
Current1[ConversionCount] = AdcRegs.ADCRESULT0 >>4;
Current2[ConversionCount] = AdcRegs.ADCRESULT1 >>4;
Voltage1[ConversionCount] = AdcRegs.ADCRESULT2 >>4;
Voltage2[ConversionCount] = AdcRegs.ADCRESULT3 >>4;
Voltage3[ConversionCount] = AdcRegs.ADCRESULT4 >>4;
Voltage4[ConversionCount] = AdcRegs.ADCRESULT5 >>4;
Voltage5[ConversionCount] = AdcRegs.ADCRESULT6 >>4;
Voltage6[ConversionCount] = AdcRegs.ADCRESULT7 >>4;
```

در کد زیر، جریان I_{p_real} همان جریان دشارژ سلول با بیشترین شارژ در یکسان‌سازی غیرفعال (یا همان جریان اولیه ترانس در یکسان‌سازی فعال) است و جریان I_{s_real} همان جریان ثانویه ترانس در یکسان‌سازی فعال است. در مرحله اول ($TIMER_COUNT=0$)، سلول ماکزیمم در مقاومت شنت یا اولیه ترانس دشارژ می‌شود و در مرحله دوم ($TIMER_COUNT=1$)، شارژ ذخیره شده در ثانویه ترانس به داخل استک باتری برمی‌گردد.

```
if (TIMER_COUNT==0)
{
    I1=I2=I3=I4=I5=I6=0;
    if (Max_Bat==1) I1=-0.001*Ip_real;
    if (Max_Bat==2) I2=-0.001*Ip_real;
    if (Max_Bat==3) I3=-0.001*Ip_real;
    if (Max_Bat==4) I4=-0.001*Ip_real;
    if (Max_Bat==5) I5=-0.001*Ip_real;
    if (Max_Bat==6) I6=-0.001*Ip_real;
}
if (TIMER_COUNT==1)
{
    I1=0.001*Is_real;
    I2=0.001*Is_real;
    I3=0.001*Is_real;
    I4=0.001*Is_real;
    I5=0.001*Is_real;
    I6=0.001*Is_real;
}
```

در کد زیر، ولتاژهای خوانده شده توسط ADC، به مقادیر واقعی ولتاژ هر سلول تبدیل می‌شود. ضریب تبدیل، همان بهره مدار آپامپی است که در مسیر از باتری تا ADC قرار داده شده است.

```
VB1=Voltage1[ConversionCount]*1.247;
V1=0.001*VB1;
VB2=Voltage2[ConversionCount]*1.246;
V2=0.001*VB2;
VB3=Voltage3[ConversionCount]*1.235;
V3=0.001*VB3;
VB4=Voltage4[ConversionCount]*1.257;
V4=0.001*VB4;
VB5=Voltage5[ConversionCount]*1.235;
V5=0.001*VB5;
VB6=Voltage6[ConversionCount]*1.244;
V6=0.001*VB6;
```

کد زیر، به منظور اجرای الگوریتم تخمین SOC است که از یکی از الگوریتم‌های SMO یا EKF استفاده می‌گردد. سپس مقادیر تخمینی از SOC هر سلول بر حسب میلی واحد تبدیل می‌شود. دستور GpioDataRegs.GPATOGGLE.bit.GPIO14 به منظور خاموش/روشن کردن یک LED با استفاده از پین شماره ۱۴ از پورت IO است.

```
EALLOW;
GpioDataRegs.GPATOGGLE.bit.GPIO14=1;
EDIS;
//states_Ich_update_SOCbalancing_EKF();
states_Ich_update_SOCbalancing_SMO();

SOC1   = SOCh1*1000;
SOC2   = SOCh2*1000;
SOC3   = SOCh3*1000;
SOC4   = SOCh4*1000;
SOC5   = SOCh5*1000;
SOC6   = SOCh6*1000;

EALLOW;
GpioDataRegs.GPATOGGLE.bit.GPIO14=1;
EDIS;
```

در کد زیر، با اعلام یک وقفه توسط تایمر CPU، برنامه وارد این وقفه شده و یک سری عملیات را انجام می‌دهد. کد زیر بخشی از وقفه تایمر را نشان می‌دهد که در آن سلول با بیشترین SOC (Max_Bat) و کمترین SOC (Min_Bat) را تعیین کرده و سوئیچ مربوط به سلول ماکزیمم را برای دشارژ روشن می‌کند (PINout). متغیر CpuTimer0.InterruptCount شمارنده تایمر بوده که با توجه به اینکه تایمر هر 500 ms یک وقفه اعلام می‌کند این شمارنده یکی اضافه می‌شود. متغیر TIMER_COUNT باقیمانده این شمارنده بر ۲ است.

```
interrupt void cpu_timer0_isr(void)
{
    CpuTimer0.InterruptCount++;
    TIMER_COUNT=(CpuTimer0.InterruptCount%2);
    if (TIMER_COUNT==1)
    {
        EALLOW;
        GpioDataRegs.GPACLEAR.all=0xFFFF;
        EDIS;
    }
    //***** SOC Balancing *****
    SOCmax=SOC1;
    SOCmin=SOC1;
    Max_Bat=1;
    Min_Bat=1;
    PINout=0x00000001 | 0x00000002;

    if (SOC2>SOCmax)
    {
        SOCmax=SOC2;
        Max_Bat=2;
        PINout=0x00000004 | 0x00000008;
```

```
}  
if (SOC2<SOCmin)  
{  
    SOCmin=SOC2;  
    Min_Bat=2;  
}  
if (SOC3>SOCmax)  
{  
    SOCmax=SOC3;  
    Max_Bat=3;  
    PINout=0x00000010 | 0x00000020;  
}  
if (SOC3<SOCmin)  
{  
    SOCmin=SOC3;  
    Min_Bat=3;  
}  
if (SOC4>SOCmax)  
{  
    SOCmax=SOC4;  
    Max_Bat=4;  
    PINout=0x00000040 | 0x00000080;  
}  
if (SOC4<SOCmin)  
{  
    SOCmin=SOC4;  
    Min_Bat=4;  
}  
if (SOC5>SOCmax)  
{  
    SOCmax=SOC5;  
    Max_Bat=5;  
    PINout=0x00000100 | 0x00000200;  
}  
if (SOC5<SOCmin)  
{  
    SOCmin=SOC5;  
    Min_Bat=5;  
}  
if (SOC6>SOCmax)  
{  
    SOCmax=SOC6;  
    Max_Bat=6;  
    PINout=0x00000400 | 0x00000800;  
}  
if (SOC6<SOCmin)  
{  
    SOCmin=SOC6;  
    Min_Bat=6;  
}  
}
```

در کد زیر، با استفاده از تفاضل سلول‌های با بیشترین و کمترین SOC، شرط توقف برنامه مشخص می‌شود که در اینجا مقدار 0,01 (۱۰ میلی واحد) برای این تفاضل تنظیم شده است. در صورت برآورده شده شرط توقف، چهار عدد ۱۵۰ به پورت سریال ارسال شده و برنامه متوقف می‌شود (ESTOP0).

```
if (TIMER_COUNT==0)
{
    if ((SOCmax-SOCmin)>10)
    {
        EALLOW;
        GpioDataRegs.GPASET.all=PINout;
        ConversionCount=0;
        EDIS;
    }
    else
    {
        EALLOW;
        GpioDataRegs.GPACLEAR.all=0xFFFF;
        EDIS;
        sprintf(str, "%d\t%d\t%d\t%d\n\r", 150, 150, 150, 150);
        scia_msg(str);
        ESTOP0;
    }
}
```

هر ۲۰۰۰ بار که تایمر وارد وقفه می‌شود (یعنی هر ۱ ثانیه)، یک بار مقادیر SOC و ولتاژهای ۶ سلول به پورت سریال ارسال می‌شود و شمارنده تایمر صفر می‌شود.

```
if (CpuTimer0.InterruptCount>=2000)
{
    //*****
    //***** Sent to Serial *****
    //*****

    disp1=SOCh1*1000;
    disp2=VB1;
    disp3=Vh1*1000;
    disp4=SOCh2*1000;
    disp5=VB2;
    disp6=Vh2*1000;
    disp7=SOCh3*1000;
    disp8=VB3;
    disp9=Vh3*1000;
    disp10=SOCh4*1000;
    disp11=VB4;
    disp12=Vh4*1000;
    disp13=SOCh5*1000;
    disp14=VB5;
    disp15=Vh5*1000;
    disp16=SOCh6*1000;
    disp17=VB6;
    disp18=Vh6*1000;
    sprintf(str, "%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\n\r", disp1, disp2, disp3,
    disp4, disp5, disp6, disp7, disp8, disp9);
}
```

```
scia_msg(str);
sprintf(str, "%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\n\r", disp10, disp11,
disp12, disp13, disp14, disp15, disp16, disp17, disp18);
scia_msg(str);

CpuTimer0.InterruptCount=0;
}
```

در ادامه برنامه، برخی توابع مربوط به عملکرد LCD نوشته شده است که از آوردن جزئیات کد آن‌ها در اینجا صرفنظر شده و فقط به ذکر نام آن بسنده شده است.

```
void initLCD(void)
void LCD_WriteChar(char c)
void LCD_WriteString(char * string)
void LCD_Cmd(unsigned int cmd)
```

در ادامه، نحوه پیاده‌سازی الگوریتم تخمین SMO در داخل DSP تشریح شده است. کد زیر مربوط به تخمین حالت‌های سلول شماره ۱ (Xh1) است. متغیرهای این سلول در جدول ۳-۵ آورده شده است. برای اطلاع بیشتر در مورد نحوه استخراج این رابطه می‌توانید به گزارش فاز ۲ مراجعه کنید.

```
void states_Ich_update_SOCbalancing_SMO(void)
{
    //***** SMO Equations *****
    //***** 1st battery states update*****
    Xh1[0] = SOCh1;
    Xh1[1] = Vh11;
    Xh1[2] = Xh1[2]; //Vh1-y0;
    Xh1[3] = Vh12;

    ey=V1-Vh1;
    v_ey = 2/PI*atan(w*ey);
    for (i=0;i<=3;i++)
    {
        Temp_Xh[i]=Ad[i][0]*Xh1[0]+Ad[i][1]*Xh1[1]+Ad[i][2]*Xh1[2]+Ad[i][3]*Xh1[3]+dt*B
        B[i]*I1+dt*HH[i]*ey+dt*h*TT[i]*v_ey;
    }
    for (i=0;i<=3;i++) Xh1[i]=Temp_Xh[i];
}
```

جدول ۳-۵- متغیرهای سلول شماره ۱ در الگوریتم SMO

متغیر	توصیف متغیر	متغیر	توصیف متغیر
V1	ولتاژ ترمینال سلول	I1	جریان گذرنده از سلول
SOCh1	تخمین SOC	Vh1	تخمین ولتاژ ترمینال سلول
Vh11	تخمین ولتاژ شبکه RC اول	Vh12	تخمین ولتاژ شبکه RC دوم

کد مربوط به تخمین حالت‌های بقیه سلول‌ها با SMO در زیر آمده است که عیناً مشابه سلول شماره ۱ است.

```

//***** 1st battery states update*****
Xh2[0] = SOCh2;
Xh2[1] = Vh21;
Xh2[2] = Xh2[2]; //Vh2-y0;
Xh2[3] = Vh22;

ey=V2-Vh2;
v_ey = 2/PI*atan(W*ey);
for (i=0;i<=3;i++)
{

    Temp_Xh[i]=Ad[i][0]*Xh2[0]+Ad[i][1]*Xh2[1]+Ad[i][2]*Xh2[2]+Ad[i][3]*Xh2[3]+dt*B
B[i]*I2+dt*HH[i]*ey+dt*h*TT[i]*v_ey;
}
for (i=0;i<=3;i++) Xh2[i]=Temp_Xh[i];

//***** 1st battery states update*****
Xh3[0] = SOCh3;
Xh3[1] = Vh31;
Xh3[2] = Xh3[2]; //Vh3-y0;
Xh3[3] = Vh32;

ey=V3-Vh3;
v_ey = 2/PI*atan(W*ey);
for (i=0;i<=3;i++)
{

    Temp_Xh[i]=Ad[i][0]*Xh3[0]+Ad[i][1]*Xh3[1]+Ad[i][2]*Xh3[2]+Ad[i][3]*Xh3[3]+dt*B
B[i]*I3+dt*HH[i]*ey+dt*h*TT[i]*v_ey;
}
for (i=0;i<=3;i++) Xh3[i]=Temp_Xh[i];

//***** 1st battery states update*****
Xh4[0] = SOCh4;
Xh4[1] = Vh41;
Xh4[2] = Xh4[2]; //Vh4-y0;
Xh4[3] = Vh42;

ey=V4-Vh4;
v_ey = 2/PI*atan(W*ey);
for (i=0;i<=3;i++)
{

    Temp_Xh[i]=Ad[i][0]*Xh4[0]+Ad[i][1]*Xh4[1]+Ad[i][2]*Xh4[2]+Ad[i][3]*Xh4[3]+dt*B
B[i]*I4+dt*HH[i]*ey+dt*h*TT[i]*v_ey;
}
for (i=0;i<=3;i++) Xh4[i]=Temp_Xh[i];

//***** 1st battery states update*****
Xh5[0] = SOCh5;
Xh5[1] = Vh51;
Xh5[2] = Xh5[2]; //Vh5-y0;
Xh5[3] = Vh52;

```

```

ey=V5-Vh5;
v_ey = 2/PI*atan(W*ey);
for (i=0;i<=3;i++)
{
    Temp_Xh[i]=Ad[i][0]*Xh5[0]+Ad[i][1]*Xh5[1]+Ad[i][2]*Xh5[2]+Ad[i][3]*Xh5[3]+dt*B
B[i]*I5+dt*HH[i]*ey+dt*h*TT[i]*v_ey;
}
for (i=0;i<=3;i++) Xh5[i]=Temp_Xh[i];

//***** 1st battery states update*****
Xh6[0] = SOCh6;
Xh6[1] = Vh61;
Xh6[2] = Xh6[2]; //Vh6-y0;
Xh6[3] = Vh62;

ey=V6-Vh6;
v_ey = 2/PI*atan(W*ey);
for (i=0;i<=3;i++)
{
    Temp_Xh[i]=Ad[i][0]*Xh6[0]+Ad[i][1]*Xh6[1]+Ad[i][2]*Xh6[2]+Ad[i][3]*Xh6[3]+dt*B
B[i]*I6+dt*HH[i]*ey+dt*h*TT[i]*v_ey;
}
for (i=0;i<=3;i++) Xh6[i]=Temp_Xh[i];

//***** All battery states update*****
SOCh1 = Xh1[0];
Vh11 = Xh1[1];
Vh1 = Xh1[2]+y0;
Vh12 = Xh1[3];

SOCh2 = Xh2[0];
Vh21 = Xh2[1];
Vh2 = Xh2[2]+y0;
Vh22 = Xh2[3];

SOCh3 = Xh3[0];
Vh31 = Xh3[1];
Vh3 = Xh3[2]+y0;
Vh32 = Xh3[3];

SOCh4 = Xh4[0];
Vh41 = Xh4[1];
Vh4 = Xh4[2]+y0;
Vh42 = Xh4[3];

SOCh5 = Xh5[0];
Vh51 = Xh5[1];
Vh5 = Xh5[2]+y0;
Vh52 = Xh5[3];

```



```
SOCh6 = Xh6[0];
Vh61 = Xh6[1];
Vh6 = Xh6[2]+y0;
Vh62 = Xh6[3];
}
```

کد زیر، مربوط به پیاده‌سازی الگوریتم تخمین EKF در داخل DSP برای تخمین حالت‌های سلول شماره ۱ (Xh1) است. متغیرهای این سلول در جدول ۶-۳ آورده شده است. برای اطلاع بیشتر در مورد نحوه استخراج این رابطه می‌توانید به گزارش فاز ۲ مراجعه کنید.

```

//*****
void states_Ich_update_SOChbalancing_EKF(void)
{
    //***** EKF Equations *****
    //***** 1st battery states update*****
    Xh1[0]=SOCh1;
    Xh1[1]=Vh11;
    Xh1[2]=Vh12;
    H[0]=a1+2*a2*SOCh1+a3*(pow(SOCh1,2))-b*a4*exp(-b*SOCh1);
    K_temp=H[0]*(H[0]*P1[0][0]+H[1]*P1[1][0]+H[2]*P1[2][0])+H[1]*(H[0]*P1[0][1]+H[1]*P1[1][1]+H[2]*P1[2][1])+H[2]*(H[0]*P1[0][2]+H[1]*P1[1][2]+H[2]*P1[2][2])+R;
    for (i=0;i<=2;i++)
    {
        K[i]=(P1[i][0]*H[0]+P1[i][1]*H[1]+P1[i][2]*H[2])/K_temp;
    }
    OCV = a0+a1*SOCh1+a2*(pow(SOCh1,2))+a3*(pow(SOCh1,3))+a4*exp(-b*SOCh1);
    Vh1 = OCV+Vh11+Vh12+R0*I1;
    for (i=0;i<=2;i++)
    {
        Temp_Xh[i]=A[i][0]*Xh1[0]+A[i][1]*Xh1[1]+A[i][2]*Xh1[2]+B[i]*I1+(A[i][0]*K[0]+A[i][1]*K[1]+A[i][2]*K[2])*(V1-Vh1);
        //Temp_Xh[i]=0;
        for (j=0;j<=2;j++)
        {
            // Temp_Xh[i]=Temp_Xh[i]+A[i][j]*Xh1[j]+A[i][j]*K[j]*(V1-Vh1);
            P1_Temp1[i][j]=A[i][0]*(EYE[0][j]-K[0]*H[j])+A[i][1]*(EYE[1][j]-K[1]*H[j])+A[i][2]*(EYE[2][j]-K[2]*H[j]);
            P1_Temp2[i][j]=P1[i][0]*A[j][0]+P1[i][1]*A[j][1]+P1[i][2]*A[j][2];
        }
    }
    for (i=0;i<=2;i++) Xh1[i]=Temp_Xh[i]; //+B[i]*I1;
}

```

جدول ۶-۳- متغیرهای سلول شماره ۱ در الگوریتم EKF

متغیر	توصیف متغیر	متغیر	توصیف متغیر
-------	-------------	-------	-------------

V1	ولتاژ ترمینال سلول	I1	جریان گذرنده از سلول
SOCh1	تخمین SOC	Vh1	تخمین ولتاژ ترمینال سلول
Vh11	تخمین ولتاژ شبکه RC اول	Vh12	تخمین ولتاژ شبکه RC دوم
Xh1	بردار حالت سلول شماره ۱	H[i], i=0,1,2	مشتق خروجی نسبت به حالت Xh1[i]

کد مربوط به تخمین حالت‌های بقیه سلول‌ها با EKF در زیر آمده است که عیناً مشابه سلول شماره ۱ است.

```
//***** 2nd battery states update*****
Xh2[0]=SOCh2;
Xh2[1]=Vh21;
Xh2[2]=Vh22;
H[0]=a1+2*a2*SOCh2+3*a3*(pow(SOCh2,2))-b*a4*exp(-b*SOCh2);
K_temp=H[0]*(H[0]*P2[0][0]+H[1]*P2[1][0]+H[2]*P2[2][0])+H[1]*(H[0]*P2[0][1]+H[1]*P2[1][1]+H[2]*P2[2][1])+H[2]*(H[0]*P2[0][2]+H[1]*P2[1][2]+H[2]*P2[2][2])+R;
for (i=0;i<=2;i++)
{
    K[i]=(P2[i][0]*H[0]+P2[i][1]*H[1]+P2[i][2]*H[2])/K_temp;
}
OCV = a0+a1*SOCh2+a2*(pow(SOCh2,2))+a3*(pow(SOCh2,3))+a4*exp(-b*SOCh2);
Vh2 = OCV+Vh21+Vh22+R0*I2;
for (i=0;i<=2;i++)
{
    Temp_Xh[i]=A[i][0]*Xh2[0]+A[i][1]*Xh2[1]+A[i][2]*Xh2[2]+B[i]*I2+(A[i][0]*K[0]+A[i][1]*K[1]+A[i][2]*K[2])*(V2-Vh2);
    for (j=0;j<=2;j++)
    {
        P2_Temp1[i][j]=A[i][0]*(EYE[0][j]-K[0]*H[j])+A[i][1]*(EYE[1][j]-K[1]*H[j])+A[i][2]*(EYE[2][j]-K[2]*H[j]);
        P2_Temp2[i][j]=P2[i][0]*A[j][0]+P2[i][1]*A[j][1]+P2[i][2]*A[j][2];
    }
}
for (i=0;i<=2;i++)    Xh2[i]=Temp_Xh[i];

//***** 3rd battery states update*****
Xh3[0]=SOCh3;
Xh3[1]=Vh31;
Xh3[2]=Vh32;
H[0]=a1+2*a2*SOCh3+3*a3*(pow(SOCh3,2))-b*a4*exp(-b*SOCh3);
K_temp=H[0]*(H[0]*P3[0][0]+H[1]*P3[1][0]+H[2]*P3[2][0])+H[1]*(H[0]*P3[0][1]+H[1]*P3[1][1]+H[2]*P3[2][1])+H[2]*(H[0]*P3[0][2]+H[1]*P3[1][2]+H[2]*P3[2][2])+R;
for (i=0;i<=2;i++)
{
    K[i]=(P3[i][0]*H[0]+P3[i][1]*H[1]+P3[i][2]*H[2])/K_temp;
}
OCV = a0+a1*SOCh3+a2*(pow(SOCh3,2))+a3*(pow(SOCh3,3))+a4*exp(-b*SOCh3);
Vh3 = OCV+Vh31+Vh32+R0*I3;
```

```

    for (i=0;i<=2;i++)
    {
        Temp_Xh[i]=A[i][0]*Xh3[0]+A[i][1]*Xh3[1]+A[i][2]*Xh3[2]+B[i]*I3+(A[i][0]*K[0]+A[i][1]*K[1]+A[i][2]*K[2])*(V3-Vh3);
        for (j=0;j<=2;j++)
        {
            P3_Temp1[i][j]=A[i][0]*(EYE[0][j]-K[0]*H[j])+A[i][1]*(EYE[1][j]-K[1]*H[j])+A[i][2]*(EYE[2][j]-K[2]*H[j]);

            P3_Temp2[i][j]=P3[i][0]*A[j][0]+P3[i][1]*A[j][1]+P3[i][2]*A[j][2];
        }
    }
    for (i=0;i<=2;i++) Xh3[i]=Temp_Xh[i];

    //***** 4th battery states update*****
    Xh4[0]=SOCh4;
    Xh4[1]=Vh41;
    Xh4[2]=Vh42;
    H[0]=a1+2*a2*SOCh4+3*a3*(pow(SOCh4,2))-b*a4*exp(-b*SOCh4);
    K_temp=H[0]*(H[0]*P4[0][0]+H[1]*P4[1][0]+H[2]*P4[2][0])+H[1]*(H[0]*P4[0][1]+H[1]*P4[1][1]+H[2]*P4[2][1])+H[2]*(H[0]*P4[0][2]+H[1]*P4[1][2]+H[2]*P4[2][2])+R;
    for (i=0;i<=2;i++)
    {
        K[i]=(P4[i][0]*H[0]+P4[i][1]*H[1]+P4[i][2]*H[2])/K_temp;
    }
    OCV = a0+a1*SOCh4+a2*(pow(SOCh4,2))+a3*(pow(SOCh4,3))+a4*exp(-b*SOCh4);
    Vh4 = OCV+Vh41+Vh42+R0*I4;
    for (i=0;i<=2;i++)
    {
        Temp_Xh[i]=A[i][0]*Xh4[0]+A[i][1]*Xh4[1]+A[i][2]*Xh4[2]+B[i]*I4+(A[i][0]*K[0]+A[i][1]*K[1]+A[i][2]*K[2])*(V4-Vh4);
        for (j=0;j<=2;j++)
        {
            P4_Temp1[i][j]=A[i][0]*(EYE[0][j]-K[0]*H[j])+A[i][1]*(EYE[1][j]-K[1]*H[j])+A[i][2]*(EYE[2][j]-K[2]*H[j]);

            P4_Temp2[i][j]=P4[i][0]*A[j][0]+P4[i][1]*A[j][1]+P4[i][2]*A[j][2];
        }
    }
    for (i=0;i<=2;i++) Xh4[i]=Temp_Xh[i];

    //***** 5th battery states update*****
    Xh5[0]=SOCh5;
    Xh5[1]=Vh51;
    Xh5[2]=Vh52;
    H[0]=a1+2*a2*SOCh5+3*a3*(pow(SOCh5,2))-b*a4*exp(-b*SOCh5);
    K_temp=H[0]*(H[0]*P5[0][0]+H[1]*P5[1][0]+H[2]*P5[2][0])+H[1]*(H[0]*P5[0][1]+H[1]*P5[1][1]+H[2]*P5[2][1])+H[2]*(H[0]*P5[0][2]+H[1]*P5[1][2]+H[2]*P5[2][2])+R;
    for (i=0;i<=2;i++)
    {

```

```

        K[i]=(P5[i][0]*H[0]+P5[i][1]*H[1]+P5[i][2]*H[2])/K_temp;
    }
    OCV = a0+a1*SOCh5+a2*(pow(SOCh5,2))+a3*(pow(SOCh5,3))+a4*exp(-b*SOCh5);
    Vh5 = OCV+Vh51+Vh52+R0*I5;
    for (i=0;i<=2;i++)
    {
        Temp_Xh[i]=A[i][0]*Xh5[0]+A[i][1]*Xh5[1]+A[i][2]*Xh5[2]+B[i]*I5+(A[i][0]*K[0]+A[i][1]*K[1]+A[i][2]*K[2])*(V5-Vh5);
        for (j=0;j<=2;j++)
        {
            P5_Temp1[i][j]=A[i][0]*(EYE[0][j]-K[0]*H[j])+A[i][1]*(EYE[1][j]-K[1]*H[j])+A[i][2]*(EYE[2][j]-K[2]*H[j]);

            P5_Temp2[i][j]=P5[i][0]*A[j][0]+P5[i][1]*A[j][1]+P5[i][2]*A[j][2];
        }
        for (i=0;i<=2;i++) Xh5[i]=Temp_Xh[i];

        //***** 6th battery states update*****
        Xh6[0]=SOCh6;
        Xh6[1]=Vh61;
        Xh6[2]=Vh62;
        H[0]=a1+2*a2*SOCh6+3*a3*(pow(SOCh6,2))-b*a4*exp(-b*SOCh6);
        K_temp=H[0]*(H[0]*P6[0][0]+H[1]*P6[1][0]+H[2]*P6[2][0])+H[1]*(H[0]*P6[0][1]+H[1]*P6[1][1]+H[2]*P6[2][1])+H[2]*(H[0]*P6[0][2]+H[1]*P6[1][2]+H[2]*P6[2][2])+R;
        for (i=0;i<=2;i++)
        {
            K[i]=(P6[i][0]*H[0]+P6[i][1]*H[1]+P6[i][2]*H[2])/K_temp;
        }
        OCV = a0+a1*SOCh6+a2*(pow(SOCh6,2))+a3*(pow(SOCh6,3))+a4*exp(-b*SOCh6);
        Vh6 = OCV+Vh61+Vh62+R0*I6;
        for (i=0;i<=2;i++)
        {
            Temp_Xh[i]=A[i][0]*Xh6[0]+A[i][1]*Xh6[1]+A[i][2]*Xh6[2]+B[i]*I6+(A[i][0]*K[0]+A[i][1]*K[1]+A[i][2]*K[2])*(V6-Vh6);
            for (j=0;j<=2;j++)
            {
                P6_Temp1[i][j]=A[i][0]*(EYE[0][j]-K[0]*H[j])+A[i][1]*(EYE[1][j]-K[1]*H[j])+A[i][2]*(EYE[2][j]-K[2]*H[j]);

                P6_Temp2[i][j]=P6[i][0]*A[j][0]+P6[i][1]*A[j][1]+P6[i][2]*A[j][2];
            }
        }
        for (i=0;i<=2;i++) Xh6[i]=Temp_Xh[i];

        //***** All battery states update*****
        for (i=0;i<=2;i++)
        {
            for (j=0;j<=2;j++)
            {

```

```

        P1[i][j]=P1_Temp1[i][0]*P1_Temp2[0][j]+P1_Temp1[i][1]*P1_Temp2[1][j]+P1_Temp1[i][2]*P1_Temp2[2][j]+Q[i][j];

        P2[i][j]=P2_Temp1[i][0]*P2_Temp2[0][j]+P2_Temp1[i][1]*P2_Temp2[1][j]+P2_Temp1[i][2]*P2_Temp2[2][j]+Q[i][j];

        P3[i][j]=P3_Temp1[i][0]*P3_Temp2[0][j]+P3_Temp1[i][1]*P3_Temp2[1][j]+P3_Temp1[i][2]*P3_Temp2[2][j]+Q[i][j];

        P4[i][j]=P4_Temp1[i][0]*P4_Temp2[0][j]+P4_Temp1[i][1]*P4_Temp2[1][j]+P4_Temp1[i][2]*P4_Temp2[2][j]+Q[i][j];

        P5[i][j]=P5_Temp1[i][0]*P5_Temp2[0][j]+P5_Temp1[i][1]*P5_Temp2[1][j]+P5_Temp1[i][2]*P5_Temp2[2][j]+Q[i][j];

        P6[i][j]=P6_Temp1[i][0]*P6_Temp2[0][j]+P6_Temp1[i][1]*P6_Temp2[1][j]+P6_Temp1[i][2]*P6_Temp2[2][j]+Q[i][j];
    }
}

SOCh1 = Xh1[0];
Vh11 = Xh1[1];
Vh12 = Xh1[2];

SOCh2 = Xh2[0];
Vh21 = Xh2[1];
Vh22 = Xh2[2];

SOCh3 = Xh3[0];
Vh31 = Xh3[1];
Vh32 = Xh3[2];

SOCh4 = Xh4[0];
Vh41 = Xh4[1];
Vh42 = Xh4[2];

SOCh5 = Xh5[0];
Vh51 = Xh5[1];
Vh52 = Xh5[2];

SOCh6 = Xh6[0];
Vh61 = Xh6[1];
Vh62 = Xh6[2];
}

```

در ادامه برنامه، برخی توابع مربوط به عملکرد پورت سریال (SCIA) نوشته شده است که از آوردن جزئیات کد آن‌ها در اینجا صرفنظر شده و فقط به ذکر نام آن بسنده کرده‌ایم.

```

void scia_echoback_init()
void scia_xmit(int a) // Transmit a character from the SCI
void scia_msg(char * msg)

```

```
void scia_fifo_init() // Initialize the SCI FIFO
```

کد زیر مربوط به راه‌اندازی پورت‌های IO می‌باشد. در اینجا فقط کد مربوط به دو پین از پورت IO برای نمونه آورده شده است.

```
void init_Gpio(void)
{
    EALLOW;
    GpioCtrlRegs.GPAMUX1.bit.GPIO0 = 0;
    GpioCtrlRegs.GPADIR.bit.GPIO0 = 1;
    GpioDataRegs.GPACLEAR.bit.GPIO0=1;

    GpioCtrlRegs.GPAMUX1.bit.GPIO1 = 0;
    GpioCtrlRegs.GPADIR.bit.GPIO1 = 1;
    GpioDataRegs.GPACLEAR.bit.GPIO1=1;
}
```

۵-۳-۳- لحاظ کردن مدیریت دمایی

یکی از نگرانی‌های اساسی در زمینه ایمنی باتری‌های لیتیم-یون، آسیب‌پذیری این باتری‌ها در دمای بالا است. دمای بالا ممکن است باعث تحریک شدن واکنش‌های به شدت گرمازا شوند که دما را بیشتر افزایش می‌دهد [۹]. چنین فرایندی که در دمایی فراتر از آستانه دمایی معینی اتفاق می‌افتد به تدریج می‌تواند منجر به احتراق شود.

دما نه تنها روی ولتاژ مدار باز باتری (OCV)، بلکه روی ظرفیت باتری نیز تأثیر می‌گذارد [۱۰]. ظرفیت نامی باتری معمولاً در دمای اتاق یعنی 25°C اندازه‌گیری می‌شود. ظرفیت موجود در باتری با کاهش دما کاهش می‌یابد و هنگامی که دما به زیر 20°C افت می‌کند می‌تواند نصف شود. دمای باتری همچنین روی انرژی که می‌تواند از باتری کشیده شود تأثیرگذار است. ظرفیت باتری در دماهای بالاتر معمولاً بیشتر از ظرفیتش در دماهای پایین‌تر است. اما بالا بردن دمای باتری روش مؤثری برای افزایش ظرفیت آن نیست، چرا که این کار باعث کاهش عمر مفید باتری خواهد شد. دما همچنین روی مقاومت داخلی باتری نیز تأثیرگذار است. هنگامی که دمای داخل یک باتری افزایش یابد، الکترون‌ها تحریک شده و بنابراین، مقاومت داخلی آن کاهش خواهد یافت و باتری می‌تواند جریان بیشتری تولید کند. از طرفی دیگر، اساساً هنگامی که دما به زیر 10°C افت پیدا می‌کند، هم ظرفیت انرژی و هم ظرفیت توان باتری‌های لیتیم-یون تنزل می‌یابد [۱۱].

در کاربردهای خودروی برقی/هیبریدی (EV/HEV)، یک مدل گرمایی از باتری که بتواند دمای باتری را در شرایط مختلف شارژ و دشارژ پیش‌بینی کند، برای توسعه الگوریتم‌های مدیریت گرمایی و استراتژی‌های خنک‌سازی ضروری است. از آنجایی که اختلاف دمای بین سلول‌ها روی عدم توازن، قابلیت اطمینان و عملکرد سلول اثر دارد مدیریت گرمایی سلول‌ها ضروری است. دمای عملکردی باتری‌های لیتیم-یون 45°C است، بنابراین دمای باتری‌های لیتیم-یون باید مانیتور شود تا از این حد آستانه تجاوز نکند.

۴-۳- جمع بندی

در این فصل به تشریح مدار استخراج داده‌های تست شناسایی و مدار یکسان‌سازی شارژ فعال و غیرفعال بر روی پردازشگر DSP و همچنین کدهای نرم‌افزاری مربوط به هر یک از مدارهای فوق پرداخته شد. مدار استخراج داده تست شناسایی از یک پردازنده AVR جهت اجرای سیکل‌های دشارژ و نمونه‌برداری از جریان و ولتاژ باتری در طول فرایند دشارژ بهره می‌برد. نمونه‌های اندازه‌گیری شده به صورت برخط بر روی درگاه سریال دستگاه قرار می‌گیرد و از طریق اتصال این درگاه با یک کامپیوتر می‌توان این داده‌ها را ذخیره نمود. این داده‌ها سپس در کامپیوتر و در نرم‌افزار Matlab تحلیل شده و پارامترهای مدل باتری استخراج می‌گردند.

از آنجایی که SOC به طور مستقیم قابل اندازه‌گیری نیست باید تخمین زده شود. برای تخمین SOC از دو روش رؤیتگر مد لغزشی (SMO) و فیلتر کالمن توسعه یافته (EKF) استفاده شده است. در واقع، هر یک از روش‌های یکسان‌سازی شارژ غیرفعال و فعال، یک بار با استفاده از الگوریتم SMO و بار دیگر با استفاده از EKF پیاده‌سازی شده است.

همچنین در این فصل کد پیاده‌سازی روش‌های یکسان‌سازی شارژ غیرفعال و فعال بر روی پردازشگر DSP تشریح گردید و بعد از معرفی اجزای تشکیل‌دهنده مدار یکسان‌سازی شارژ، به تشریح کدهای مربوط به تخمینگر EKF و SMO پرداخته شده است.

۵-۳- مراجع

- [1] M. Chen, and G. A. Rincon-Mora, "Accurate electrical battery model capable of predicting runtime and I-V performance," IEEE Trans. Energy Convers., vol. 21, no. 2, pp. 504-511, Jun. 2006.
- [2] N. A. Chaturvedi, R. Klein, J. Christensen, J. Ahmed, and A. Kojic, "Algorithms for Advanced Battery-Management Systems: Modeling, estimation, and control challenges for lithium-ion battery," IEEE Control Syst. Mag., vol. 30, no. 3, pp. 49-68, Jun. 2010.
- [3] A. R. Pishrobat, "Estimating the parameters of charge, discharge and state of health of the battery using identification and nonlinear estimation techniques, MSc. Thesis, University of Tehran, electrical and Computer School of Engineering, Sep. 2007.
- [4] R. J. Grube, Automotive Battery State-of-Health Monitoring Methods, M.Sc. Thesis, Wright State University, Dec. 2008.
- [5] B. T. Kuhn, G. E. Pitel, and P. T. Krein, "Electrical properties and equalization of lithium-ion cells in automotive applications," in *Vehicle Power and Propulsion IEEE Conference*, 2005.
- [6] J. Zhao, J. Jiang, and L. Niu, "A novel charge equalization technique for electric vehicle battery system," *International Conference on Power Electronics and Drive Systems*, vol. 2, pp. 853-857, 2003.
- [7] M. Gholizadeh, F.R. Salmasi, "Estimation of state of charge, unknown nonlinearities, and state of health of a lithium-ion battery based on a comprehensive unobservable model," IEEE Transaction on Industrial Electronics, Vol. 61, No. 3, march 2014.
- [8] J. W. Shin, G. S. Seo, C.Y. Chun, and B. H. Cho, "Selective flyback balancing circuit with improved balancing speed for series connected Lithium-ion batteries," in *Power Electronics Conference (IPEC)*, pp. 1180-1184, 2010.
- [9] T. M. Bandhauer, S. Garimella, and T. F. Fuller, "A critical review of thermal issues in lithium-ion batteries," Journal of The Electrochemical Society, vol. 158, pp. 1-25, 2011.
- [10] D. Doerffel and S. A. Sharkh, "A critical review of using the peukert equation for determining the remaining capacity of lead-acid and lithium-ion batteries," Journal of Power Sources, vol. 155, no. 2, pp. 395-400, 2006.
- [11] S. Zhang, K. Xu, and T. Jow, "A new approach toward improved low temperature performance of lithium battery," Electrochemistry communications, vol. 4, no. 11, pp. 928-932, 2002.

۴- انجام آزمایش و ارزیابی نتایج

۴-۱- مقدمه

فرآیند مدیریت شارژ باتری شامل ۲ پروسه مجزا می باشد: ۱- شناسایی پارامترهای دینامیکی باتری و ۲- متعادل کردن شارژ سلول های باتری با استفاده از پارامترهای شناسایی شده. برای شناسایی پارامترها نیاز به اجرای آزمایشات جامعی است که عموماً ساعت ها طول می کشد. به همین دلیل عموماً شناسایی به صورت آفلاین صورت می گیرد. به این معنی که ابتدا یکی از سلول های باتری به ترمینال مربوط به شناسایی وصل شده و با اجرای آزمایشات شناسایی، و تغییرات برنامه ریزی شده بار، ولتاژ و جریان دو سر سلول اندازه گیری و ذخیره می گردد. سپس با تحلیل داده های اندازه گیری شده به صورت نرم افزاری پارامترهای باتری استخراج می گردند. چنانچه سلول ها تماماً از یک مدل و یک برند باشند می توان فرض کرد که همگی از پارامترهای مشابهی برخوردارند و پارامترهای شناسایی شده را به باقی سلول ها تعمیم داد.

پس از شناسایی پارامترهای دینامیکی باتری، می توان فرآیند متعادل کردن شارژ را با استفاده از مدل های دینامیکی مورد نظر و پیش بینی رفتار باتری تحت بارهای مختلف، انجام داد. به این منظور همانطور که در گزارشات قبلی گفته شده، تخمینی از سطح شارژ سلول های باتری با استفاده از یکی از روش فیلتر کالمن توسعه یافته و کنترل مد لغزشی بدست می آید. با داشتن میزان سطح شارژ تمامی سلول ها در هر لحظه، شارژ سلول دارای بیشترین شارژ به صورت فعال بر روی باقی سلول ها پخش می شود و یا به صورت غیرفعال توسط یک مقاومت اتلاف می شود. این فرآیند ادامه می یابد تا جایی که شارژ تمامی سلول ها با یک تفرانس مشخص با یکدیگر برابر شوند.

در این فصل به راه اندازی و اجرای سیستم مدیریت شارژ باتری توسعه داده شده و انجام پروسه های شناسایی و متعادل کردن شارژ پرداخته می شود. به این منظور در ابتدا سیستم مدیریت باتری از نظر سخت افزاری، تجهیزات و متعلقات تشریح می گردد. در ادامه با اجرای آزمایشات شناسایی، نتایج استخراج پارامترها به تفصیل بررسی می گردد. سپس به اجرای پروسه یکسان سازی شارژ از طریق اتصال سلول ها به دستگاه و راه اندازی نرم افزاری پرداخته می شود.

۲-۴- معرفی دستگاه یکسانساز شارژ

دستگاه ساخته شده برای یکسانسازی شارژ استک باتری، از دو بخش اصلی مجزا تشکیل شده است: بخش شناسایی و بخش یکسانساز. هر کدام از این بخش‌ها شامل یک یا چند برد الکترونیکی می‌باشد که همگی در داخل یک محفظه قرار گرفته و ورودی‌ها و خروجی‌های لازم برای دسترسی به این بردها بر روی محفظه تعبیه شده است. این دستگاه در زیر نشان داده شده است.



شکل ۱-۴- دستگاه یکسانساز شارژ

شکل ۲-۴ نمای جلو و پشت این باکس را نشان می‌دهد. همانطور که مشاهده می‌کنید در جلوی باکس یک LCD کاراکتری و یک سری LED، کلید و کانکتور دوتایی تعبیه شده است. نمایشگر LCD به منظور نشان دادن سطح شارژ و ولتاژ ۶ سلول تحت بالانس قرار داده شده است. کلید ON/OFF برای خاموش/روشن کردن دستگاه است. کلید A/P نوع یکسانسازی فعال/غیرفعال را تعیین می‌کند که البته این کار در داخل کد انجام خواهد شد. کلید Fan برای خاموش/روشن کردن فن خنک کننده در پشت دستگاه است. همچنین چراغهایی برای نشان دادن وضعیت فعلی دستگاه بر روی آن قرار گرفته است که Active/Passive بودن نوع یکسانسازی و همچنین نوع متغیری که قرار است در باتری‌ها بالانس شود (SOC/Voltage) را نشان می‌دهند. چراغ Power و Fan به ترتیب وضعیت خاموش/روشن بودن دستگاه و فن را نشان می‌دهند. چراغ Temp نیز برای مانیتور کردن وضعیت دمایی به کار می‌رود.

کانکتور BI جهت وصل کردن یک سلول باتری به آن به منظور شناسایی پارامتر، مورد استفاده قرار می‌گیرد. سر مثبت باتری باید به کانکتور قرمز و سر منفی به کانکتور مشکی وصل شود. همچنین ۶ کانکتور دیگر برای اتصال ۶ سلول باتری به منظور یکسانسازی شارژ این سلول‌ها می‌باشد.

در پشت باکس نیز یک سری کانکتور/پورت تعبیه شده است. کانکتور Power برای تغذیه دستگاه است که به برق ۲۲۰ ولت شهر وصل می‌شود. پورت USB برای اتصال Emulator (پروگرامر DSP) قرار گرفته در داخل محفظه) به کامپیوتر از طریق یک کابل دو سر USB تعبیه شده است. همچنین دو کانکتور DB9 با برچسب‌های AVR و DSP نیز در پشت دستگاه تعبیه شده است که اولی برای اتصال سریال برد شناسایی و

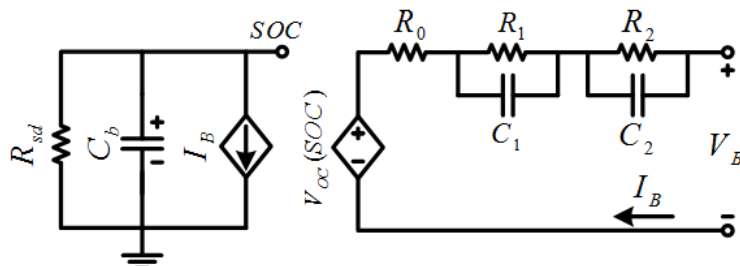
دومی برای اتصال سریال برد DSP به کامپیوتر جهت ارسال داده‌های باتری مانند ولتاژ یا سطح شارژ به کامپیوتر به کار می‌رود. این پورت‌ها باید از طریق مبدل سریال به USB به کامپیوتر وصل شود. در ادامه این فصل، نحوه استفاده از هر کدام از بردهای شناسایی و یکسانسازی تشریح خواهد شد.



شکل ۲-۴- نمای جلوی محفظه (شکل بالا) و نمای پشت (شکل پایین)

۳-۴- شناسایی پارامترهای باتری

در این بخش به شناسایی پارامترهای مدل یک سلول باتری پرداخته خواهد شد. مدل انتخابی باتری، مدل رندل مرتبه دوم می‌باشد که پیش‌تر در فصل دوم از گزارش معرفی شد [۵]. این مدل در شکل ۳-۴ آمده است.



شکل ۳-۴- مدل الکتریکی باتری

مقاومت R_{sd} مقاومت اهمی باتری و شبکه‌های RC برای مدل کردن رفتار پاسخ گذاری ولتاژ باتری است. R_{sd} بیانگر مقاومت دشارژ خودبه‌خودی و C_b بیانگر خازن ظرفیت کامل است که C_n همان ظرفیت آمپرساعت باتری است. منبع ولتاژ کنترل شده در این مدل، ولتاژ مدار باز باتری V_{OC} است که تابعی غیرخطی از SOC می‌باشد و به صورت زیر مدل می‌شود [۱].

$$V_{OC}(SOC) = \alpha_0 + \alpha_1 SOC + \alpha_2 SOC^2 + \alpha_3 SOC^3 + \alpha_4 e^{-\beta SOC} \quad (4-1)$$

که در آن $SOC \in [0,1]$ و $\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \beta$ پارامترهای ثابت هستند. V_B و I_B به ترتیب ولتاژ ترمینال و جریان ترمینال باتری را نشان می‌دهند. ولتاژ پایانه باتری به صورت زیر بیان می‌شود:

$$V_B = V_{OC}(SOC) - R_0 I_B - V_1 - V_2 \quad (4-2)$$

از سمت چپ مدل، SOC به صورت زیر بیان می‌شود:

$$\dot{SOC} = -\frac{1}{R_{sd}C_b} SOC - \frac{1}{C_b} I_B \quad (4-3)$$

روابط مربوط به ولتاژهای دو سر شبکه‌های RC به صورت زیر به دست می‌آید:

$$\begin{aligned} \dot{V}_1 &= -\frac{1}{R_1 C_1} V_1 + \frac{1}{C_1} I_B \\ \dot{V}_2 &= -\frac{1}{R_2 C_2} V_2 + \frac{1}{C_2} I_B \end{aligned} \quad (4-4)$$

۴-۳-۱- روند شناسایی پارامترهای باتری

جهت اجرای پروسه شناسایی پارامترهای باتری با استفاده از برد شناسایی، دو سر یک سلول باتری را به کانکتور BI تعبیه شده در جلوی باکس وصل می‌کنیم. سر مثبت باتری باید به کانکتور قرمز و سر منفی به کانکتور مشکی وصل شود. حال پورت سریال مربوط به AVR در پشت دستگاه را از طریق یک مبدل سریال به USB به کامپیوتر وصل می‌کنیم. با اتصال پاور دستگاه در پشت باکس، برد AVR شروع به کار کرده و داده ولتاژ-جریان باتری را بر روی پورت سریال ارسال می‌کند.

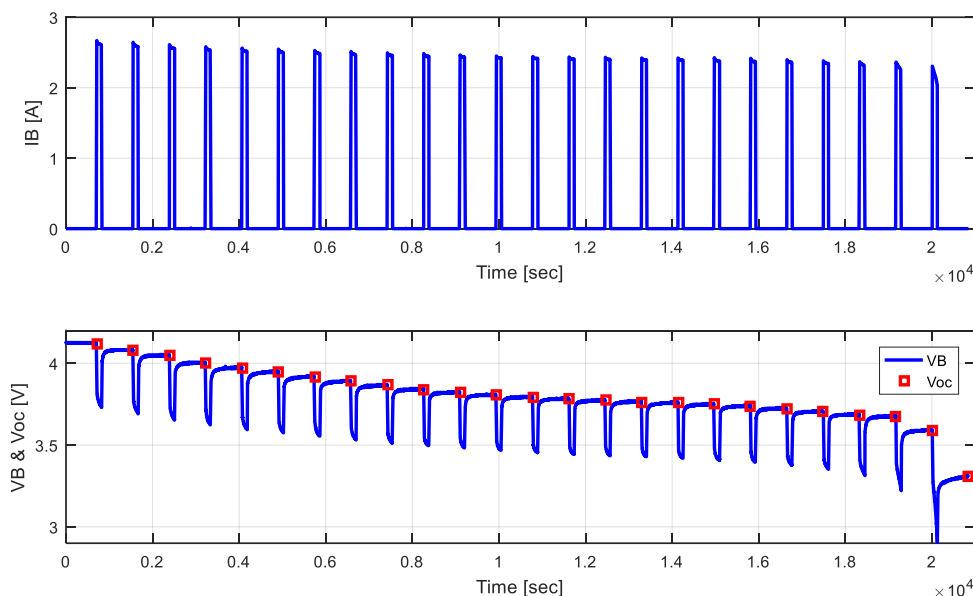
برای خواندن داده‌های باتری در کامپیوتر می‌توان از هر نرم‌افزار خواندن از پورت سریال استفاده کرد. در این پروژه از نرم‌افزار MATLAB برای این منظور استفاده شده است. یک برنامه با نام read_serial نوشته شده که با اجرای آن در متلب از پورت سریال می‌خواند و در فایل‌های با پسوند mat ذخیره می‌کند. بعد از اتمام دشارژ باتری (رسیدن به انتهای دشارژ که ولتاژ ۳ ولت می‌باشد)، مدار شناسایی قطع شده و باز تعبیه شده بر روی آن برای چند ثانیه به صدا در می‌آید. همچنین برنامه دیگری در متلب با نام Serial_Data_Merge بلافاصله بعد از این زمان، فراخوانی شده و فایل‌های mat را با هم ادغام می‌کند و یک فایل با نام test_data تولید می‌شود. سپس برنامه Modify_TestData فراخوانی می‌شود و یک فایل نهایی با نام test_data_modeified تولید می‌کند که علاوه بر داده ولتاژ-جریان، داده ولتاژ مدار باز (OCV) و سطح شارژ (SOC) باتری نیز در آن قرار دارد. این فایل شامل داده‌هایی است که برای شناسایی پارامترهای باتری استفاده می‌شود.

در نهایت با اجرای برنامه estimation_BatteryParam که از فایل test_data_modeified در آن استفاده شده است، پارامترهای مدل باتری شناسایی می‌شود. در فصل دوم از گزارش فاز سوم پروژه، روابط

ریاضی مربوط به مدل باتری و پارامترهایی که باید شناسایی شود و نحوه استخراج داده‌های ولتاژ مدار باز، مقاومت داخلی باتری و سطح شارژ باتری به طور کامل به همراه نتایج تست عملی آورده شده است.

۲-۳-۴- استخراج منحنی V_{OC} بر حسب SOC

ولتاژ مدار باز باتری، V_{OC} ، تابعی غیرخطی از SOC است. طبق تعریف، ولتاژ مدار باز باتری ولتاژ اندازه‌گیری شده در ترمینال باتری است هنگامی که هیچ باری به آن وصل نباشد و تمام فرایندهای شیمیایی داخلی آن به حالت دائمی رسیده باشند. برای استخراج منحنی V_{OC} بر حسب SOC، یک تست دشارژ برای باتری طراحی شده است که در آن یک بار مقاومتی با مقاومت نامی ۱.۴ اهم به باتری اعمال می‌شود و باتری از حالت شارژ کامل 4.1 V ($SOC=1$) با نرخ 1.33 C شروع به دشارژ می‌کند. ابتدا باتری به مدت ۷۲۰ ثانیه استراحت کرده، سپس به مدت ۱۲۰ ثانیه دشارژ می‌شود و این روند ادامه پیدا می‌کند تا باتری تا ولتاژ قطع 3 V دشارژ شود. ولتاژ و جریان اندازه‌گیری شده در طول دشارژ در شکل ۴-۴ نشان داده شده است. همان‌طور که در شکل مشاهده می‌شود، ولتاژ انتهای زمان استراحت به عنوان ولتاژ مدار باز باتری در نظر گرفته شده است.

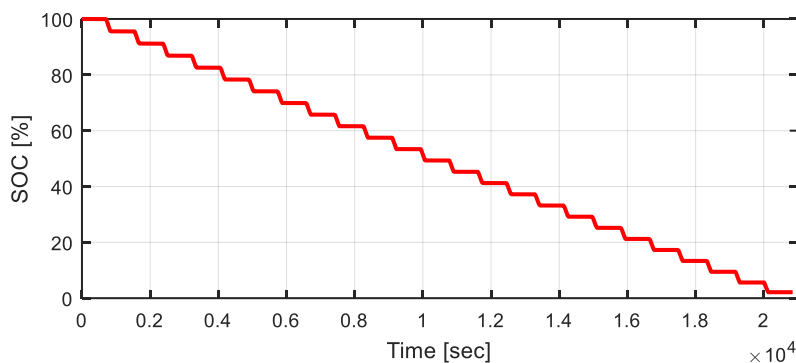


شکل ۴-۴- ولتاژ و جریان اندازه‌گیری شده به همراه نقاط مربوط به V_{OC}

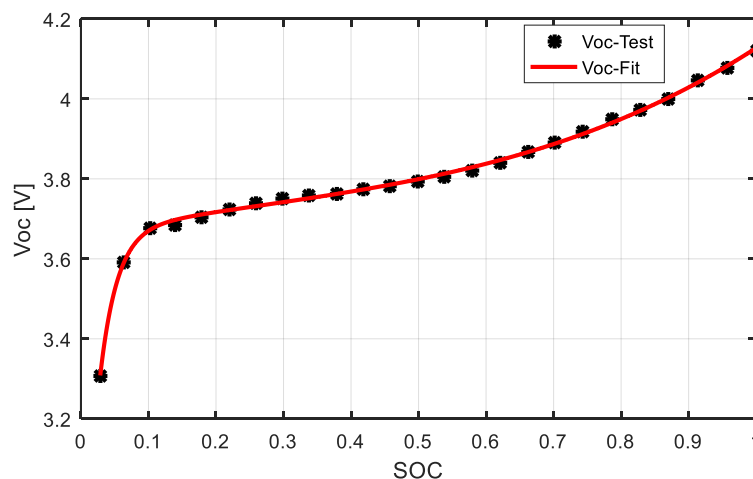
از روی جریان باتری می‌توان SOC را به روش شمارش آمپرساعت (Ah Counting) محاسبه کرد. SOC محاسبه شده در شکل ۴-۵ آمده است. حال با داشتن مقادیر V_{OC} و نیز SOC، منحنی V_{OC} بر حسب SOC به صورت شکل ۴-۶ می‌تواند رسم شود. با برآزش رابطه (۴-۱) بر منحنی V_{OC} بر حسب SOC، پارامترهای این منحنی به صورت زیر به دست می‌آیند.

$$V_{OC}(SOC) = \alpha_0 + \alpha_1 SOC + \alpha_2 SOC^2 + \alpha_3 SOC^3 + \alpha_4 e^{-\beta SOC} \quad (4-5)$$

$$\alpha_0 = 3.661, \alpha_1 = 0.333, \alpha_2 = -0.360, \alpha_3 = 0.4933, \alpha_4 = -1.122, \beta = 39.622$$



شکل ۴-۵ محاسبه شده باتری به روش شمارش آمپرساعت



شکل ۴-۶ منحنی Voc بر حسب SOC

۳-۳-۴- تخمین پارامترها

برای تخمین پارامترهای مدل باتری می‌توان از داده‌های همین تست استفاده کرد. خازن ظرفیت کامل C_b موجود در سمت چپ مدل باتری با استفاده از رابطه $C_b = 3600C_n$ برابر با ۷۲۰۰ فاراد قرار داده می‌شود که $C_n = 2$ آمپرساعت نامی باتری است.

مقاومت دشارژ خودبخودی (R_{sd}) طوری به دست آمده است که SOC به دست آمده از رابطه (۳-۴) بر SOC به دست آمده از روش شمارش کلمب منطبق باشد. برای این مقاومت مقدار ۲۰۰ اهم به دست آمده است.

برای محاسبه R_0 از نسبت تغییرات ولتاژ ناشی از پرش جریان استفاده می‌کنیم [۸]. مقدار R_0 در i امین نقطه پرش جریان می‌تواند به صورت زیر بیان شود:

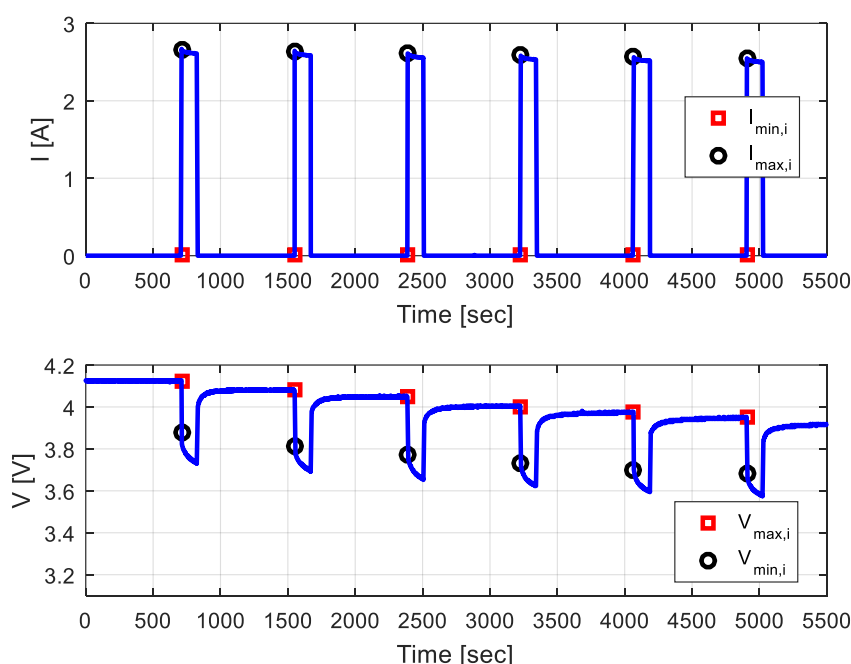
$$R_{0,i} = \frac{v_{\max,i} - v_{\min,i}}{i_{\max,i} - i_{\min,i}} \quad (4-6)$$

که $v_{\max,i}$, $v_{\min,i}$, $i_{\max,i}$, $i_{\min,i}$ در شکل ۴-۷ نشان داده شده است. مقدار میانگین $R_{0,i}$ در چند نقطه پرش جریان، R_0 را می‌دهد:

$$R_0 = \sum_{i=1}^n R_{0,i} / n \quad (4-7)$$

برای محاسبه پارامترهای شبکه‌های RC، از ترکیب روابط (۴-۲) و (۴-۴)، فرم لاپلاس رابطه (۴-۲) به صورت زیر به دست می‌آید:

$$\begin{aligned} V_B &= V_{OC}(SOC) - \left(R_0 + \frac{R_1}{1 + R_1 C_1 s} + \frac{R_2}{1 + R_2 C_2 s} \right) I_B \\ &= V_{OC}(SOC) - R_0 I_B - G(s) I_B \end{aligned} \quad (4-8)$$



شکل ۴-۷- جریان و ولتاژ استفاده شده برای محاسبه R_0

از رابطه (۴-۸) داریم:

$$V_{RC}(t) = G(s) I_B(t) = -V_B(t) + V_{OC}(SOC) - R_0 I_B(t) \quad (4-9)$$

با فرض $dI_B / dt \approx 0$ در یک زمان نمونه‌برداری، رابطه فوق به صورت زیر گسسته می‌شود:

$$V_{RC}[k] = \left(\frac{R_1(1 - e^{-T_s/T_{p1}})z^{-1}}{1 - e^{-T_s/T_{p1}}z^{-1}} + \frac{R_2(1 - e^{-T_s/T_{p2}})z^{-1}}{1 - e^{-T_s/T_{p2}}z^{-1}} \right) I_B[k] = \left(\frac{z^{-1}(b_1 + b_2 z^{-1})}{1 + a_1 z^{-1} + a_2 z^{-2}} \right) I_B[k] \quad (4-10)$$

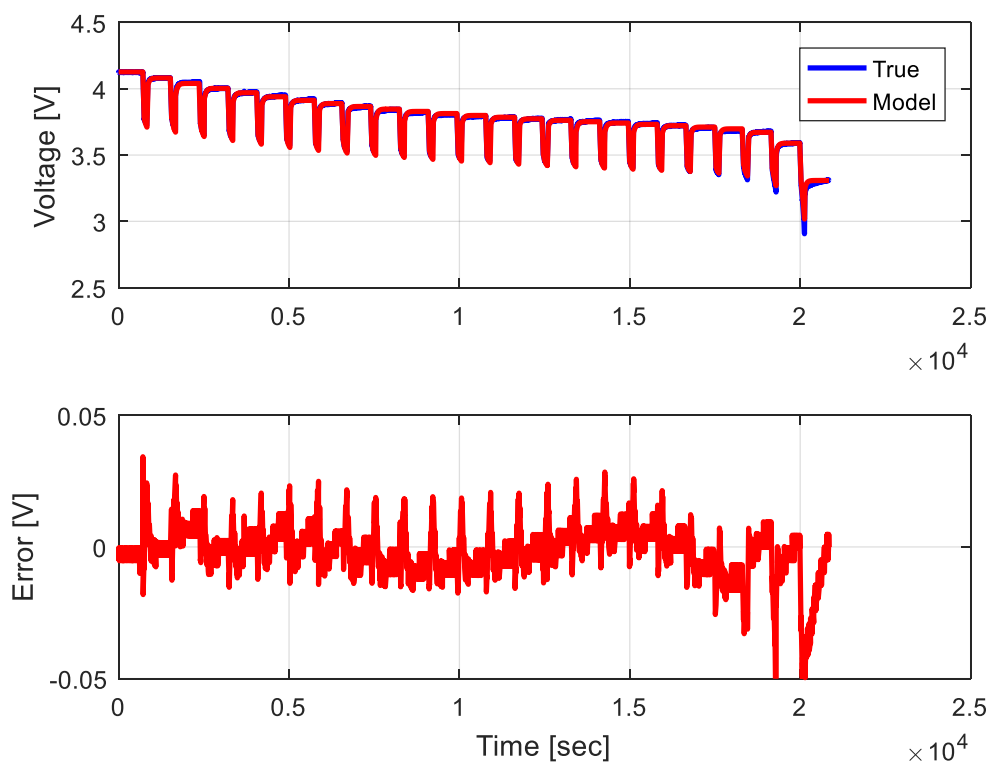
که در آن T_s زمان نمونه‌برداری است. پارامترهای a_2, a_1, b_2, b_1 با استفاده از مدل دینامیکی ARMAX تخمین زده می‌شوند و از روی آن‌ها پارامترهای شبکه‌های RC به دست می‌آیند. آنچه که در اینجا لازم به ذکر است، نحوه انتخاب ورودی، غنی بودن ورودی و انتخاب فرکانس آن است. بررسی‌های صورت گرفته در حین پیاده‌سازی روش‌های پیشنهادی نشان داده است که ورودی پالس به منظور شناسایی تمام پارامترهای مدل باتری یک ورودی غنی می‌باشد. فرکانس این ورودی باید طوری انتخاب شود که دوره تناوب آن از کوچک‌ترین ثابت زمانی سیستم کوچک‌تر باشد، در غیر این صورت پارامترهای مدل ممکن است مقادیری منفی به دست آیند.

از آنجایی که کوچک‌ترین ثابت زمانی سیستم را نداریم، فرکانس مناسب با سعی و خطا به دست می‌آید؛ مثلاً اگر در صورت انتخاب فرکانس 10Hz پارامترها منفی شد، فرکانس بزرگتری را انتخاب می‌کنیم. نتایج تخمین پارامترها در جدول ۴-۱ آورده شده است.

جدول ۴-۱- پارامترهای مدل باتری

R_0	105.5 m Ω
R_1	2.1 m Ω
R_2	37.8 m Ω
C_1	80 F
C_2	1311 F
T_{p1}	0.172 sec
T_{p2}	49.6 sec

خروجی اندازه‌گیری شده و شبیه‌سازی در شکل ۴-۸ آورده شده است. همان‌طور که در شکل دیده می‌شود حداکثر خطای مدل‌سازی برابر با 50 mV می‌باشد.



شکل ۴-۸- خروجی اندازه‌گیری شده در مقایسه با خروجی مدل

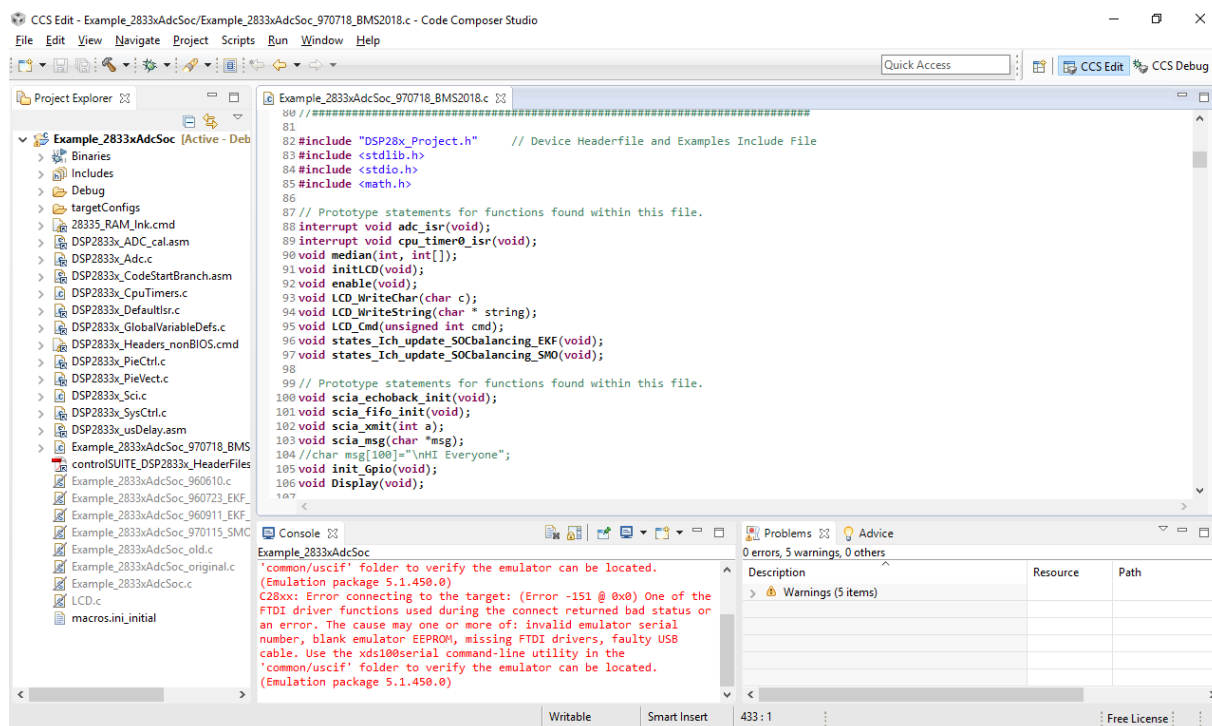
۴-۴- نتایج عملی یکسان سازی شارژ سلول های باتری

در این بخش نتایج تست های عملی یکسان سازی شارژ آورده شده است. در این پروژه چهار تست عملی برای یکسان سازی شارژ انجام شده است که عبارتند از: یکسان سازی غیرفعال با روش EKF، یکسان سازی غیرفعال با روش SMO، یکسان سازی فعال با روش EKF، یکسان سازی فعال با روش SMO. برای انتخاب روش تخمین SMO یا EKF کافی است در داخل برنامه وقفه ADC، تابع مربوط به هر تخمینگر را انتخاب کنیم. تخمینگرهای EKF و SMO به ترتیب در توابع `states_Ich_update_SOCbalancing_EKF` و `states_Ich_update_SOCbalancing_SMO` پیاده سازی شده است. بعد از انجام این تنظیمات، DSP را با استفاده از Emulator برنامه ریزی می کنیم.

همچنین برای انتخاب روش یکسان سازی فعال و غیرفعال، کفایت سوئیچ IGBT مربوط به آن روش را روشن کنیم که برای این منظور یک سوئیچ بر روی مدار قرار داده شده است که با اتصال آن به زمین، روش غیرفعال و با اتصال آن به $V_{CC}=5V$ ، روش فعال انتخاب خواهد شد. با انجام این تنظیمات و اتصال تغذیه مدار (ولتاژ ۲۴ V)، یکسان سازی شروع خواهد شد. ولتاژ و SOC هر باتری برای ذخیره سازی به داخل کامپیوتر ارسال می گردد. در ادامه به بررسی نتایج اجرای روش های یکسان سازی می پردازیم.

۴-۴-۱- نحوه استفاده از مدار یکسان ساز شارژ

جهت استفاده از مدار یکسان ساز، دو سر شش سلول باتری را به کانکتورهای B1 تا B6 تعبیه شده در جلوی باکس وصل می کنیم. سر مثبت هر سلول باید به کانکتور قرمز و سر منفی به کانکتور مشکی وصل شود. سپس پورت USB تعبیه شده در پشت باکس را از طریق یک کابل دو سر USB و پورت سریال مربوط به DSP در پشت دستگاه را از طریق یک مبدل سریال به USB به کامپیوتر وصل می کنیم. حال پاور دستگاه در پشت باکس را وصل می کنیم. برخلاف برد AVR که با اتصال پاور شروع به کار می کند، برد DSP باید از طریق نرم افزار راه اندازی شود. برای این منظور، کد نوشته شده در داخل DSP را با نرم افزار CodeComposer باز می کنیم (شکل ۹-۴).



شکل ۹-۴- برنامه DSP در نرم‌افزار CodeComposer

قبل از اجرای برنامه لازم است ۳ متغیر در داخل برنامه مقداردهی شود که عبارتند از: ۱- متغیر ActPass_Select که مقدار 0 یا 1 به ترتیب برای انتخاب نوع یکسانسازی پسیو یا اکتیو است، ۲- متغیر SOC_Voltage_Select که مقدار 0 یا 1 به ترتیب برای یکسانسازی ولتاژ یا SOC باتری‌ها است، ۳- متغیر Fan_Status که با انتخاب 0 یا 1 به ترتیب وضعیت خاموش یا روشن بودن فن را تعیین می‌کند. وضعیت این متغیرها با چراغ‌هایی بر روی باکس مشخص می‌شود. متغیرهای زیادی در داخل برنامه مورد استفاده قرار گرفته است که در در فصل سوم از گزارش فاز سوم پروژه تشریح گردیده است.

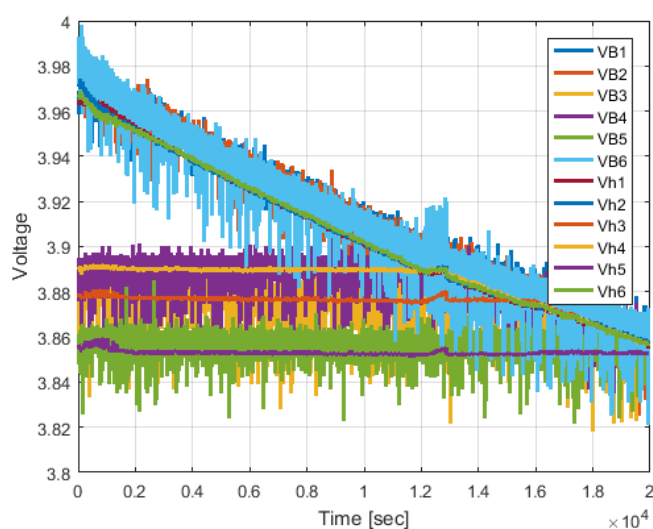
برای اجرای برنامه، ابتدا بر روی دکمه Debug کلیک می‌کنیم. اگر کد هیچ خطایی نداشته باشد و همچنین اگر کابل امولاتور (Emulator) به درستی وصل شده باشد، برنامه وارد مد اجرا می‌شود. با کلیک بر روی دکمه Run برد DSP شروع به کار می‌کند و داده‌های ولتاژ و SOC سلول‌های باتری را بر روی پورت سریال ارسال می‌کند. داده‌های ولتاژ اندازه‌گیری شده، ولتاژ تخمین زده شده و SOC هر سلول (در مجموع ۱۸ مقدار) بر روی پورت سریال به کامپیوتر ارسال می‌گردد.

برای خواندن داده‌های باتری در کامپیوتر از نرم‌افزار MATLAB استفاده شده است. یک برنامه با نام read_serial_dsp نوشته شده که با اجرای آن در متلب از پورت سریال می‌خواند و در فایل‌های با پسوند .mat ذخیره می‌کند. بعد از اتمام یکسانسازی (که در اینجا شرط یکسانسازی اختلاف حداقل و حداکثر SOC کمتر از ۱٪ است)، مدار یکسانسازی قطع می‌شود. همچنین برنامه دیگری در متلب با نام Serial_Data_Merge بلافاصله بعد از قطع مدار، فراخوانی شده و فایل‌های mat را با هم ادغام می‌کند و یک فایل با نام test_data تولید می‌شود. سپس برنامه Modify_TestData فراخوانی می‌شود و سیگنال سطح شارژ و ولتاژ سلول‌های

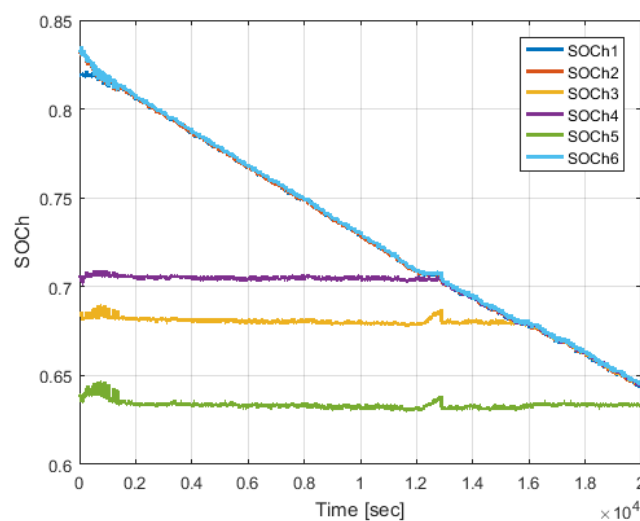
باتری ترسیم می‌گردد. در فصل چهارم از گزارش فاز سوم پروژه، نتایج یکسانسازی ۶ سلول سری باتری در دو نوع مختلف فعال و غیرفعال و همچنین با دو روش تخمین EKF و SMO آورده شده است.

۲-۴-۴- یکسان سازی غیرفعال با EKF

در اینجا با قرار دادن یک مقاومت بر سر راه سلول با بیشترین سطح شارژ، این سلول شروع به دشارژ می‌کند تا سطح شارژ آن کاهش یابد. مقاومت دشارژ برابر با ۲ اهم انتخاب شده است تا سلول‌ها با جریانی نزدیک به جریان نامی خود دشارژ شوند. نتایج یکسان سازی غیرفعال با روش EKF در ادامه آورده شده است. یکسان سازی سطح شارژ تا جایی ادامه می‌یابد که سطح شارژ همه سلول‌ها به سلول با کمترین سطح شارژ برسد.



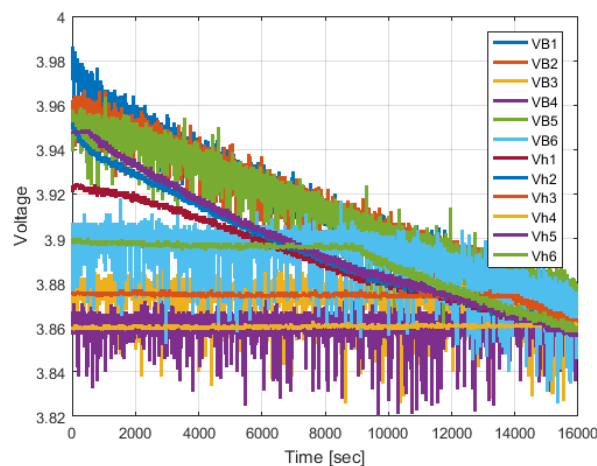
شکل ۱۰-۴- ولتاژ سلول‌ها در یکسان سازی غیرفعال با EKF



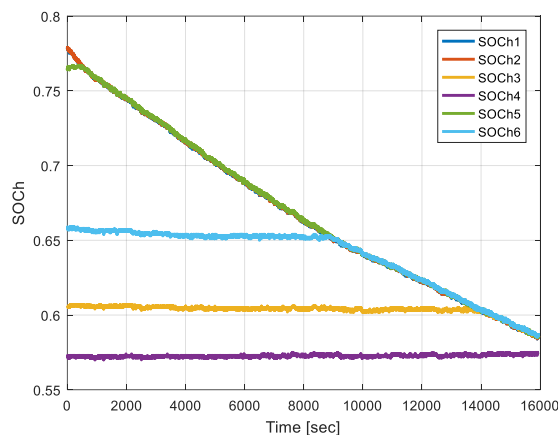
شکل ۱۱-۴- سطح شارژ سلول‌ها در یکسان سازی غیرفعال با EKF

۳-۴-۴- یکسان سازی غیرفعال با SMO

در اینجا نیز مشابه حالت قبل، یکسان سازی با قرار دادن یک مقاومت بر سر راه سلول با بیشترین سطح شارژ انجام می‌شود، با این تفاوت که تخمین سطح شارژ با روش SMO انجام می‌شود. نتایج یکسان سازی غیرفعال با روش SMO در ادامه آورده شده است.



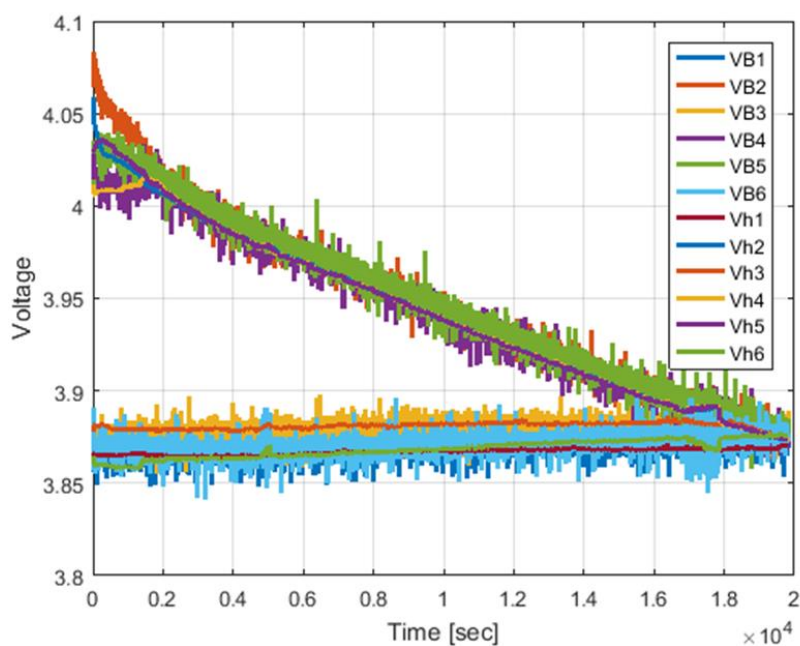
شکل ۴-۱۲- ولتاژ سلول‌ها در یکسان سازی غیرفعال با SMO



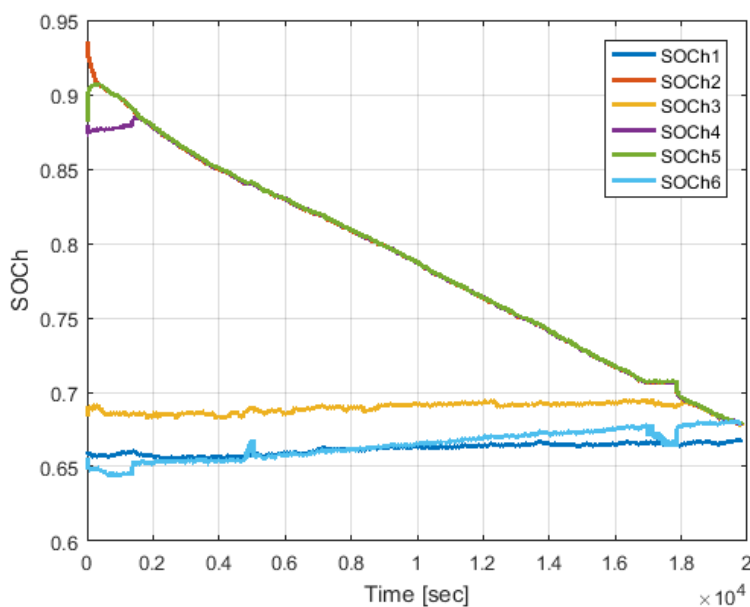
شکل ۴-۱۳- سطح شارژ سلول‌ها در یکسان سازی غیرفعال با SMO

۴-۴-۴- یکسان سازی فعال با EKF

در اینجا بعد از اینکه سلول با بیشترین سطح شارژ مشخص شد، شارژ اضافی آن در داخل یک سلف (سیم پیچ اولیه ترانس) تخلیه می‌شود تا جایی که جریان سلف به یک مقدار حداکثر از پیش تعیین شده برسد. سپس این شارژ از طریق سیم پیچ ثانویه ترانس به داخل سلول‌ها بازگردانده می‌شود (سلول‌ها را شارژ می‌کند) تا جایی که جریان سیم پیچ به صفر برسد. بعد از این مرحله، بار دیگر سلول با بیشترین سطح شارژ انتخاب شده و مرحله قبل تکرار می‌شود. یکسان سازی سطح شارژ تا جایی ادامه می‌یابد که اختلاف بیشترین و کمترین سطح شارژ سلول‌ها به حدود ۱٪ برسد. نتایج یکسان سازی فعال با EKF در ادامه آورده شده است.



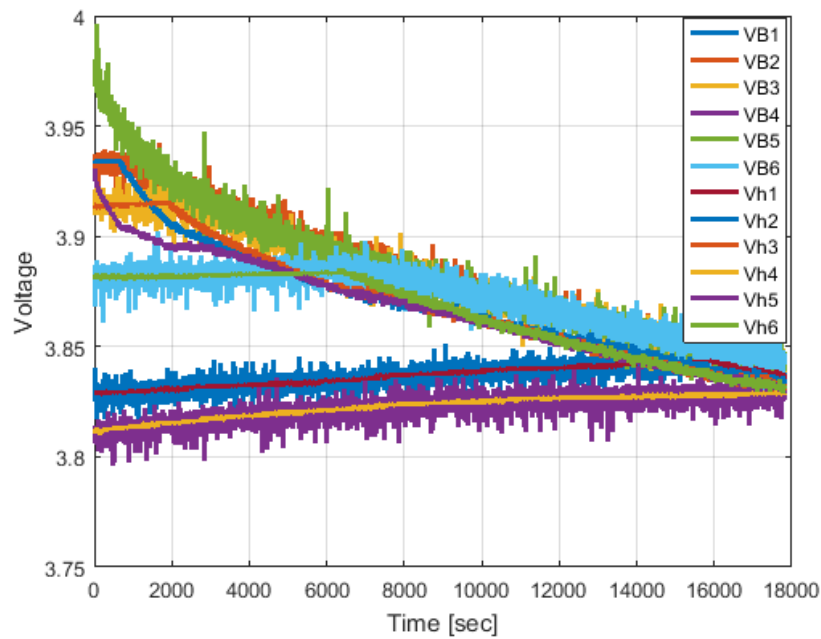
شکل ۱۴-۴- ولتاژ سلول‌ها در یکسان‌سازی فعال با EKF



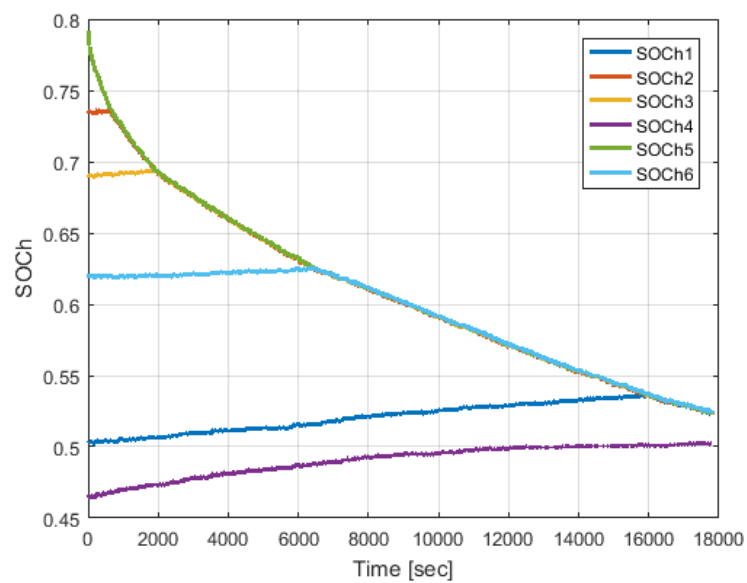
شکل ۱۵-۴- سطح شارژ سلول‌ها در یکسان‌سازی فعال با EKF

۵-۴-۴- یکسان‌سازی فعال با SMO

در اینجا مشابه حالت قبل، شارژ سلول با بیشترین سطح شارژ در داخل یک سلف تخلیه می‌شود و دوباره به داخل سلول‌ها بازگردانده می‌شود، با این تفاوت که تخمین سطح شارژ با روش SMO انجام می‌شود. نتایج یکسان‌سازی فعال با SMO در ادامه آورده شده است.



شکل ۱۶-۴- ولتاژ سلول‌ها در یکسان‌سازی فعال با SMO



شکل ۱۷-۴- سطح شارژ سلول‌ها در یکسان‌سازی فعال با SMO

۵-۴- جمع‌بندی

در این فصل، نتایج مربوط به پیاده‌سازی روش‌های یکسان‌سازی شارژ فعال و غیرفعال با هر دو تخمینگر EKF و SMO توضیح داده شده است. در یکسان‌سازی غیرفعال، شارژ اضافی سلول با بیشترین شارژ در داخل یک مقاومت تلف می‌شود و این کار تا جایی ادامه می‌یابد که سطح شارژ همه سلول‌ها به سلول با کمترین سطح شارژ برسد. برای یکسان‌سازی فعال، شارژ اضافی سلول با بیشترین شارژ در داخل یک ترانس ذخیره شده و دوباره به داخل سلول‌ها بازگردانده می‌شود و این کار تا جایی ادامه می‌یابد که اختلاف بیشترین و کمترین سطح شارژ به حدود ۱٪ برسد.

در ادامه، با تشریح نحوه استخراج داده‌های مناسب برای تست از طریق اتصال کامپیوتر به درگاه سریال دستگاه، چگونگی شناسایی پارامترها با استفاده از داده‌های تولید شده به تفصیل بررسی گردید و با انجام آزمایش از صحت اجرای پروسه اطمینان حاصل شد.

در انتها با راه‌اندازی مدار یکسان‌سازی شارژ سلول‌های باتری، عملیات یکسان‌سازی شارژ در پروسه دشارژ همزمان ۴ عدد باتری لیتیوم-یون مورد آزمایش قرار گرفت و درستی اجرای این پروسه توسط دستگاه BMS ساخته شده مورد تایید قرار گرفت.

پیوست

جدول ۱-۰- مشخصات پک باتری مورد تست در پروژه

No.	Items	Specification	Remarks
1	Rated Capacity	2000 mAh	C/2 rate, 3V cut-off
2	Nominal Voltage	3.7 V	4.2V to 3V for 1Cell
3	Discharge Cut-off Voltage	3 V	
4	Charging Voltage	4.2 V	at CV mode
5	Standard Charge Current	1 A	at CC mode
6	Rapid Charge Current	2 A	at CC mode
7	End-of-Charge Current	0.1 A	at CV mode
8	Standard Discharging Current	1 A	
9	Rapid Discharge Current	2 A	
10	Cell Weight	≈ 37 gr	
11	Standard Charging Time	≤ 8.0 hour	
12	Operating Temperature	Charge: $0 \sim 45^{\circ}\text{C}$ Discharge: $-20 \sim 45^{\circ}\text{C}$	

جدول ۲-۰- مشخصات فنی محصول BMS تحویل داده شده

No.	Items	Specification	Remarks
1	Nominal Voltage	22 V	
2	Discharge Cut-off Voltage	18 V	
3	Max Voltage	25.2 V	
4	Max Allowable Discharge Current in Equilization Mode	Active: 2 A Passive: 2.5 A	
5	Max Allowable Discharge Current in Identification Mode	5 A	
6	Max Current in Primary of the Trans.	1.2 A	in Active Balancing
7	End-of-Balance SOC Criteria	$\Delta\text{SOC} \leq 1\%$	
8	End-of-Balance Voltage Criteria	$\Delta V \leq 0.02 \text{ V}$	
9	Operating Temperature	$0 \sim 45^{\circ}\text{C}$	

۶-۴- مراجع

- [1] M. Chen, and G. A. Rincon-Mora, "Accurate electrical battery model capable of predicting runtime and I-V performance," IEEE Trans. Energy Convers., vol. 21, no. 2, pp. 504-511, Jun. 2006.
- [2] N. A. Chaturvedi, R. Klein, J. Christensen, J. Ahmed, and A. Kojic, "Algorithms for Advanced Battery-Management Systems: Modeling, estimation, and control challenges for lithium-ion battery," IEEE Control Syst. Mag., vol. 30, no. 3, pp. 49-68, Jun. 2010.
- [3] A. R. Pishrobat, "Estimating the parameters of charge, discharge and state of health of the battery using identification and nonlinear estimation techniques, MSc. Thesis, University of Tehran, electrical and Computer School of Engineering, Sep. 2007.
- [4] R. J. Grube, Automotive Battery State-of-Health Monitoring Methods, M.Sc. Thesis, Wright State University, Dec. 2008.
- [5] P. Spagnol, S. Rossi, and S. M. Savaresi, "Kalman Filter SOC Estimation for Li-Ion Batteries," IEEE Int. Conf. Control Applications (CCA), pp. 587-592, Sep. 2011.
- [6] J. Zhao, J. Jiang, and L. Niu, "A novel charge equalization technique for electric vehicle battery system," *International Conference on Power Electronics and Drive Systems*, vol. 2, pp. 853-857, 2003.