پیاده‌سازی ممیز ثابت فیلتر کالمن بر روی FPGA برای تخمین فاصله و سرعت اهداف متحرک

شهاب‌الدین رحمانیان*، محمدحسین باطنی، محمد فرداد و مجدالدین نجفی
پژوهشکده اویونیک، دانشگاه صنعتی اصفهان، اصفهان، ایران

چکیده

در سامانه‌های ردیابی هدف، از فیلتر ردیابی برای تخمین پیاپی و هموار موقعیت و سرعت هدف متحرک با کمینه خطا استفاده می‌شود. در این مقاله، روشی برای طراحی و پیاده‌سازی سخت‌افزاری فیلتر کالمن در چنین کاربردی ارائه شده است. روش پیشنهادی شامل یک پیاده‌سازی ممیز ثابت فیلتر روی FPGA است که در آن سرعت اجرای الگوریتم از طریق موازی‌سازی عملیات غیر وابسته بهبود یافته است. پس از طراحی بر اساس مسأله داده‌شده، نسخه‌های ممیز شناور و ممیز ثابت فیلتر شبیه‌سازی و نسخه ممیز ثابت روی سخت‌افزار پیاده‌سازی شده است. برای ارزیابی کارایی فیلتر، داده‌های فاصله–سرعت یک هدف متحرک با مدل مناسب تولید و پس از چندی‌سازی و درآمیختن با اغتشاش به فیلتر اعمال می‌شوند. نتایج نشان می‌دهد که با انتخاب طول بیت مناسب، فیلتر پیاده‌سازی‌شده سریع و کارآمد بوده و با زمان اجرای حدود µs ۰/۴، موجب dB ۱۱ کاهش در خطای تخمین فاصله شده و عملکردی نزدیک به نمونه ممیز شناور فراهم می‌آورد.

واژگان کلیدی: فیلتر کالمن، پیاده‌سازی FPGA، ردیابی، تخمین فاصله، تخمین سرعت.

Fixed-point FPGA Implementation of a Kalman Filter for Range and Velocity Estimation of Moving Targets

Shahabuddin Rahmanian*, Mohammad Hossein Bateni, Mohammad Fardad & Majdeddin Najafi
Avionics Research Institute, Isfahan University of Technology, Isfahan, Iran

Abstract

Tracking filters are extensively used within object tracking systems in order to provide consecutive smooth estimations of position and velocity of the object with minimum error. Namely, Kalman filter and its numerous variants are widely known as simple yet effective linear tracking filters in many diverse applications. In this paper, an effective method is proposed for designing and implementation of a Kalman filter in an object tracking application. The considered tracking application implies the capability to produce a smooth and reliable output stream by the tracking filter, even in presence of different disturbing types of noise, including background or spontaneous noises, as well as disturbances with continues or discrete nature. The presented method includes a fixed-point implementation of the Kalman filter on FPGA, which targets the joint estimation of position-velocity pair of an intended object in heavy presence of noise. The execution speed of the Kalman algorithm is drastically enhanced in the proposed implementation. This enhancement is attained by emphasis on hardware implementation of every single computational block on the one hand, and through appropriate parallelization and pipelining of independent tasks within the Kalman process on the other hand. After designing the filter parameters with respect to the requirements of a given tracking problem, a floating-point model and a fixed-point hardware model of the filter are implemented using MATLAB and Xilinx System Generator, respectively.

* Corresponding author

*نویسندهٔ عهده‌دار مکاتبات

● تاریخ ارسال مقاله:۱۳۹۶/۸/۲ ● تاریخ پذیرش: ۱۳۹۸/۳/۲۹ ● تاریخ انتشار: ۱۳۹۸/۱۰/۰۷ ● نوع مطالعه: پژوهشی

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-10]

[DOI: 10.29252/jsdp.16.3.100]

In order to evaluate the performance of the filter under realistic circumstances, a set of appropriately defined scenarios are carried out. The simulations are carefully designed in order to represent the extremely harsh scenarios in which the input measurements to the filter are deeply polluted by different kinds of noises. In each simulation the position-velocity data corresponding to a moving object is generated according to an appropriate model, quantized, and contaminated by noise and fed into the filter. Performances of the Kalman filter in software version (i.e. the floating point replica) and hardware version (i.e. the fixed-point replica) are quantitatively compared in the designed scenario. Our comparison employs NMSE and maximum error values as quantitative measures, verifying the competency of our proposed fixed-point hardware implementation.

The results of our work show that, with adequate selection word length, the implemented filter is fast and efficient; it confines the algorithm execution time to 50 clock pulses, i.e. about 0.4 μs when a 125 MHz clock is used. It is also verified that our implementation reduces the position and velocity estimation errors by 11 dB and 1.2 dB, respectively. The implemented filter also confines the absolute values of maximum error in position and velocity to 10 meter and 0.7 meter/sec. in the considered scenario, which is almost resembles the performance of its floating point counterpart. The presented Kalman filter is finally implemented on Zc706 evaluation board and the amount of utilized hardware resource (FFs, LUTs, DSP48, etc.) are reported as well as the estimated power consumption of the implemented design. The paper is concluded through comparison of the proposed design with some recent works which confirms the efficacy of the presented implementation.

Keywords: Kalman filter, FPGA implementation, Tracking, Distance estimation, Velocity estimation.

۱- مقدمه

هدف اصلی بسیاری از سامانه‌های ردیابی هدف، محاسبهٔ دقیق فاصله و سرعت یک هدف متحرک است. درعمل، مقادیر فاصله و سرعت آشکارسازی‌شده به‌وسیلهٔ بخش پردازش سیستم ردیابی، دارای دقت محدود و آمیخته به خطای چندی‌سازی[1] است. به همین دلیل در بیش‌تر موارد استفاده از یک ساختار بدون تأخیر به‌منظور هموارسازی و کاهش خطای این داده‌ها ضروری است. ساختار بالا از روابط دینامیکی مفروض در مورد جسم متحرک و به‌طور خاص از رابطه خطی بین فاصله جسم متحرک و سرعت آن برای کاهش خطا و هموارسازی تخمین بهره می‌برد.

فیلتر کالمن یک تخمین‌زنندهٔ خطی بهینه است که برای تخمین سیگنال‌های آغشته به اغتشاش و یا خطا مورد استفاده قرار می‌گیرد [1]. از زمانی که این فیلتر در سال ۱۹۶۰ معرفی شد، به‌طور گسترده‌ای در کنترل مدرن، پردازش سیگنال در حوزه‌های مختلف مانند صوت، تصویر [2] و سیگنال‌های حیاتی [3]، مخابرات، ردیابی اهداف و ناوبری مورد استفاده قرار گرفته و پژوهش‌های زیادی در رابطه با آن توسط پژوهش‌گران انجام شده است. با توجه به این‌که فیلتر کالمن یک تخمین‌زنندهٔ بازگشتی مرتبه نخست است، می‌تواند برای تخمین بردار حالت نامعلوم از روی اندازه‌گیری آمیخته با اغتشاش داده‌های متوالی مرتبط با حالت سیستم به کار گرفته شود. این کار به‌صورت بازگشتی و با کمینه‌کردن معیار میانگین مربعات خطا انجام می‌شود. با توجه به حجم به‌نسبه زیاد محاسبات مورد نیاز در پیاده‌سازی فیلتر کالمن و این‌که در بسیاری مواقع لازم است، خروجی‌ها به‌صورت بی‌درنگ[2] تولید شوند، پیاده‌سازی آن بر روی پردازش‌گرهای دیجیتال همواره یک چالش بوده است. این مسأله اغلب به‌خاطر تعداد زیاد ضرب ماتریسی و به‌ویژه عملیات تقسیم مورد نیاز برای وارونه‌سازی ماتریس در هر تکرار از این الگوریتم است.

از دیگر مسائل مهم در زمینهٔ پیاده‌سازی فیلتر کالمن می‌توان به تأثیر خطای ناشی از چندی‌سازی مقادیر و گردکردن محاسبات بر روی پایداری فیلتر کالمن اشاره کرد. کاهش بار محاسباتی و زمان اجرای الگوریتم فیلتر از دیگر مواردی است که در پیاده‌سازی فیلتر کالمن باید مورد توجه قرار گیرد. برای حل برخی از چالش‌های بالا، راه‌کارهای مختلفی پیشنهاد شده است که از جمله آن می‌توان به روش‌هایی برای اصلاح الگوریتم به‌منظور اجرای سریع‌تر، استفاده از روش‌های فاکتورگیری[3] و استفاده از ساختارهای دیجیتال بهینه و معماری‌های پیچیدهٔ عددی اشاره کرد [4].

مشخص‌کردن مقدار صحیح ماتریس‌های کوواریانس اغتشاش و مقداردهی اولیه به ماتریس‌های کوواریانس از موارد کلیدی در تنظیم پارامترهای فیلتر کالمن هستند. گرچه خطاهایی که در انتخاب پارامترهای این ماتریس‌ها رخ می‌دهد، بیشتر مربوط به کمبود اطلاعات طراح از سیستم واقعی است تا خطای گردکردن، اما از بدو به‌کارگیری فیلتر کالمن در کاربردهای مختلف عملی، طراحان متوجه حساسیت بالای این فیلتر به خطای گردکردن در محاسبات ماتریس‌های کوواریانس نیز شدند. ماتریس کوواریانس به‌دلیل این خطاها،

[1] Quantization

[2] Real-time

[3] Factorization

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-10]

[DOI: 10.29252/jsdp.16.3.100]

می‌تواند تقارن خود را از دست داده و دارای مقادیر ویژهٔ منفی شود. این اتفاقات تأثیر مستقیمی روی بهرهٔ فیلتر کالمن داشته و سرعت هم‌گرایی الگوریتم تخمین را کم می‌کند.

یکی از راه‌های کاهش خطای گردکردن، استفاده از روش‌های فاکتورگیری به‌کمک تجزیهٔ فاکتور چالسکی یا چالسکی اصلاح‌شده است که با عنوان فیلتر جذر[1] نیز شناخته می‌شود [5]. استفاده از این روش‌ها موجب افزایش بار محاسباتی سیستم می‌شود؛ اما مقاومت فیلتر در برابر خطای گردکردن را به طرز قابل توجهی افزایش می‌دهد. به‌عنوان مثال در [6]، پیاده‌سازی فیلتر کالمن روی DSP ممیز ثابت ۳۲ بیتی TMS320F2812 با سه الگوریتم مختلف جذر، مورد بررسی قرار گرفته و نشان داده شده است که الگوریتم کارسون‌اشمیت بهترین مصالحه را بین کارایی و زمان اجرا دارد.

محاسبات مورد نیاز در فیلتر کالمن شامل جمع و ضرب ماتریس‌ها و وارونه‌سازی ماتریسی است و تعداد عملیات محاسباتی در هر بار اجرای فیلتر از مرتبه n^3 است که در آن n تعداد متغیرهای حالت است [7]. به‌منظور کاهش بار محاسباتی الگوریتم، پژوهش‌گران ویژگی‌های ماتریس‌های مختلف تعریف‌شده در الگوریتم مانند تقارن و تُنُک‌بودن را مورد بررسی قرار داده‌اند. به‌عنوان مثال، در [8] کاهش بار محاسباتی از مرتبه ۲/۷ نسبت به روش استاندارد برای پیاده‌سازی فیلتر کالمن توسعه‌یافته[2] مرتبه پنجم گزارش شده است. در [9] نیز روش مشابهی برای پیاده‌سازی یک فیلتر کالمن مرتبه چهار بر روی FPGA مورد استفاده قرار گرفته است.

به‌منظور کاهش بار محاسباتی، فیلتر کالمن می‌تواند به چندین فیلتر کالمن موازی تقسیم شود که دراصطلاح فیلتر کالمن چندطبقه نامیده می‌شود. در [10]، با اتخاذ چنین رویکردی برای نخستین‌بار یک تخمین زنندهٔ کالمن دوطبقه ارائه شده است. ایدهٔ اصلی شامل جداکردن فیلتر کالمن به دو فیلتر موازی است؛ در این حالت، انجام محاسبات روی دو فیلتر کالمن کوچک‌تر (به جای یک فیلتر کالمن بزرگ) انجام شده و این کار موجب کاهش حجم محاسبات می‌شود.

در [11] یک فیلتر کالمن دوطبقهٔ بهینه پیشنهاد شده است که در آن روابط ریاضی هر طبقه به‌طوردقیق معادل روابط ریاضی فیلتر کالمن معمولی هستند. در [12] این فیلتر دوطبقهٔ بهینه برای تخمین عملکرد سیستم‌های غیرخطی تعمیم یافته است؛ همچنین پیاده‌سازی‌های مؤثری از این فیلتر در [13] و [14] به‌منظور استفاده در درایوهای جریان متناوب پیشنهاد شده است.

در بعضی از پیاده‌سازی‌های فیلتر کالمن از پردازنده‌های دیجیتالی سیگنال‌ها[3] (DSP) استفاده می‌شود. DSPها گزینهٔ مناسبی برای پیاده‌سازی الگوریتم‌هایی با حجم بالای محاسبات برداری-ماتریسی هستند. در [15]، یکی از نخستین پیاده‌سازی‌های مبتنی بر DSP برای فیلتر کالمن پیشنهاد شده است. در این پیاده‌سازی، یک فیلتر مرتبه چهارم کالمن روی DSP ممیز ثابت TI TMS320C25 پیاده‌سازی شده است. زمان لازم برای هر بار اجرای الگوریتم کالمن در این پیاده‌سازی برابر ۲۸۳/۵ µs گزارش شده است. پیاده‌سازی دقیق فیلتر کالمن روی یک DSP ممیز شناور روش مؤثری برای کاهش تأثیر خطای محاسباتی گردکردن است؛ اما استفاده از این پردازش‌گرها باعث افزایش زمان اجرا و هزینهٔ سخت‌افزاری می‌شود.

به‌منظور کاهش مؤثر زمان اجرا، طراحان انتخاب دیگری جز موازی‌کردن وظایفی که می‌توانند به‌طور هم‌زمان اجرا شوند، ندارند. در همین راستا، پیاده‌سازی‌هایی برای فیلتر کالمن به فرم واحدهای پردازش دادهٔ موازی در [16] و [17] معرفی شده است. در این طراحی‌ها، ساختارهای موازی و خط لوله‌ای به‌صورت آرایه‌ای قابل گسترش از عملگرهای ساده معرفی شده‌اند که در آن‌ها انتقال داده تنها به‌صورت محلی انجام می‌شود. این ساختار به‌طور بالقوه می‌تواند برای کاربردهای با حجم محاسباتی بالا بسیار کارآمد باشد، اما پیاده‌سازی‌های کمی برای آن ارائه شده است. یک علت عمده این مسأله، لزوم پیاده‌سازی این ساختار به‌صورت ASIC[4] است که روشی پرهزینه و دارای انعطاف‌پذیری کم است.

یکی دیگر از بسترهای مناسب برای پیاده‌سازی عملی فیلتر کالمن استفاده از FPGA است. با توجه به قابلیت موازی‌سازی و انعطاف زیاد در پیاده‌سازی سخت‌افزار بر روی FPGA، دست‌یابی به کاهش قابل‌توجه در زمان اجرای فیلتر امکان‌پذیر خواهد بود [18]. نخستین پیاده‌سازی فیلتر کالمن بر روی FPGA در [19] منتشر شده است. در این مقاله، فیلتر به‌عنوان پردازندهٔ کمکی یا شتاب‌دهندهٔ سخت‌افزاری برای یک سیستم ردیابی چندهدفی مرتبهٔ چهارم پیشنهاد شده است. زمان لازم برای اجرای الگوریتم در این طرح ۰/۴ µs است که بسیار کمتر از زمان اجرای پیاده‌سازی بر روی DSP است.

[1] Root filter
[2] Extended Kalman Filter
[3] Digital Signal Processors
[4] Application-Specific Integrated Circuit

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-10]
[DOI: 10.29252/jsdp.16.3.100]

دلیل اصلی این مسأله استفاده از ساختار سخت‌افزاری تماماً موازی است. در [20]، یک پیاده‌سازی بر روی FPGA Spartan-6 از فیلتر کالمن توسعه‌یافته، برای درایو یک موتور سنکرون بدون حس‌گر پیشنهاد شده که زمان اجرای الگوریتم در آن در حدود ۲/۸ µs است.

امروزه پیاده‌سازی تماماً سخت‌افزاری یا تماماً نرم‌افزاری فیلتر کالمن تنها راه‌حل ممکن نیست و می‌توان از تراشه‌هایی که در کنار سخت‌افزار قابل برنامه‌ریزی واحدهای پردازش‌گر قوی نیز دارند، استفاده کرد. از جمله این تراشه‌ها، می‌توان به تراشه‌های سری Zynq شرکت Xilinx اشاره کرد که در کنار سخت‌افزار قابل برنامه‌ریزی دارای دو هستهٔ قوی پردازنده ARM9 است. چالش طراحی در این موارد، تفکیک بهینه وظایف الگوریتم برای پیاده‌سازی در سخت‌افزار یا نرم‌افزار است. بر همین مبنا در [20]، تفکیک سخت‌افزاری / نرم‌افزاری برای یک فیلتر کالمن توسعه‌یافته با استفاده از الگوریتم ژنتیک به‌صورت بهینه پیشنهاد شده است.

برخلاف وجود روش‌های شرح داده‌شدهٔ بالا برای پیاده‌سازی سخت‌افزاری انواع مختلف فیلتر کالمن، تنوع نسبی راهکارهای پیاده‌سازی این فیلتر، به‌ویژه بر روی FPGA، چندان زیاد نیست و معدود پیاده‌سازی‌های موجود نیز به‌طورعمومی متمرکز بر کاربردهای خاص بوده و به‌ندرت در مقالات مورد بررسی دقیق و جامع قرار گرفته است. به همین دلیل لزوم ارائه رویکردهای مناسب و کارآمد برای پیاده‌سازی فیلتر کالمن بر بستر FPGA محسوس و مورد توجه است.

در این مقاله، یک رویکرد کارآمد برای پیاده‌سازی سخت‌افزاری ممیز ثابت فیلتر کالمن بر بستر FPGA پیشنهاد شده و کارایی آن مورد بحث و تأیید قرار می‌گیرد. روال ارائه مطالب در مقاله حاضر بدین شرح است: در ادامه و در فصل دوم، ساختار کلی فیلتر کالمن مورد استفاده مشخص شده و معادلات حاکم بر فیلتر و مراحل مختلف الگوریتم تکرارشونده برای اجرای آن معرفی می‌شود. فصل سوم به پیاده‌سازی سخت‌افزاری فیلتر پرداخته و طرح پیشنهادی برای این منظور را شرح می‌دهد که شامل تمهیداتی برای بهبود کارایی فیلتر و بهینه‌سازی پیاده‌سازی آن است. در فصل چهارم، کارایی فیلتر ارائه‌شده مورد ارزیابی دقیق قرار می‌گیرد. برای این کار، از یک سو عملکرد فیلتر در یک سناریوی مناسب در محیط MATLAB شبیه‌سازی و نتایج آن به‌دقت و به شکل کمّی مورد ارزیابی و تحلیل قرار می‌گیرد و از سوی دیگر یک پیاده‌سازی سخت‌افزاری از فیلتر ممیز ثابت طراحی‌شده بر روی FPGA انجام شده و پارامترهای کارآمدی پیاده‌سازی و میزان مصرف منابع آن ارزیابی می‌شود. ارزیابی کارایی طرح پیشنهادی در فصل چهارم با مقایسه با برخی از کارهای انجام‌شده قبلی در همین زمینه تکمیل می‌شود. فصل پنجم و نهایی مقاله به جمع‌بندی بحث و نتیجه‌گیری تخصیص دارد.

۲- معادلات و ساختار فیلتر کالمن

محاسبات مورد نیاز در هر مرحله از اجرای فیلتر کالمن را می‌توان به دو مرحلهٔ پیش‌بینی و به‌هنگام‌سازی تقسیم کرد؛ در مرحلهٔ پیش‌بینی، با استفاده از اطلاعات دینامیک سیستم، پیش‌بینی اولیه‌ای از مقدار بردار حالت و ماتریس کوواریانس، از روی مقادیر قبلی آن‌ها، به‌دست می‌آید. در مرحلهٔ به‌هنگام‌سازی تخمین موجود از بردار حالت و ماتریس کوواریانس-که معرف انرژی مورد انتظار خطا در خروجی فیلتر است- با استفاده از حالت قبلی، دادهٔ جدید اندازه‌گیری شده و مقدار قبلی ماتریس کوواریانس به‌هنگام می‌شود. به‌همین‌منظور در پیاده‌سازی فیلتر کالمن، لازم است، متغیرهای بردار حالت و ماتریس کوواریانس همواره در حافظه نگهداری شوند. در ادامه متغیرها و معادلات فیلتر کالمن به صورت دقیق مورد بررسی قرار می‌گیرد.

۲-۱- معادلات فیلتر کالمن

معادلات فیلتر کالمن با استفاده از مفهوم فضای حالت بیان می‌شود. به این منظور یک مدل فضای حالت تعریف و تلاش می‌شود تا در هر اجرا تخمین بهینه و به‌هنگام‌شده‌ای از مقدار بردار حالت به‌دست آید. فیلتر کالمن با دانستن مدل سیستم، آمارگان متغیرهای آن و اندازه‌گیری همراه با اغتشاش سیگنال ورودی، قادر است حالت سیستم را با کمینه‌کردن معیار میانگین مربعات خطا پیش‌بینی کند.

در روابط مربوط به پیاده‌سازی فیلتر، پارامترها به صورت داده‌شده در جدول (۱) تعریف می‌شوند. در این جدول، پارامترهای مختلف فیلتر کالمن جهت تخمین فاصله و سرعت یک هدف متحرک با در دست داشتن اندازه‌گیری مغشوشی از فاصله و سرعت آن ارائه شده است. در روابط نوشته شده، Δt نشان‌دهندهٔ فاصلهٔ زمانی بین دو اجرای الگوریتم است. فرض شده است که نمونه‌های اندازه‌گیری‌شده از سیگنال ورودی نیز در بازه‌هایی به اندازهٔ Δt به‌صورت متوالی مشاهده می‌شوند.

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-10]

[DOI: 10.29252/jsdp.16.3.100]

(جدول-۱): پارامترهای فیلتر کالمن.

(Table-1): Kalman Filter Parameters.

تعریف	پارامتر
بردار حالت سیستم در لحظه k	$X_k = \begin{bmatrix} x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} r \\ v \end{bmatrix}$
ماتریس مدل گذار حالت	$F_k = F = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$
پارامتری برای محاسبه ماتریس گذار	$G_k = G = \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix}$
ماتریس گذار کوواریانس خطا	$Q_k = Q = GG^T\sigma_a^2 = \begin{bmatrix} \frac{\Delta t^4}{2} & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & \Delta t^2 \end{bmatrix}\sigma_a^2$
بردار اندازه‌گیری (مشاهدات ورودی)	$Z_k = HX_k + v_k$
ماتریس ضرایب اندازه‌گیری	$H_k = H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
توان اغتشاش اندازه‌گیری	$R_k = R = E[v_k v_k^T]$
مقدار اولیه بردار حالت	$\hat{X}_0 = \begin{bmatrix} \hat{r}_0 \\ \hat{v}_0 \end{bmatrix}$
مقدار اولیه ماتریس کوواریانس حالت	$P_0 = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_v^2 \end{bmatrix}$

مؤلفهٔ نخست X_k نشان‌دهندهٔ فاصله و مؤلفه دوم X_k نمایش‌دهندهٔ سرعت جسم دنبال‌شونده نسبت به مشاهده‌گر است که در فضای حالت خطی نمایش داده می‌شوند. هرچند استفاده از فیلتر کالمن برای تخمین بردار حالت در سیستم‌های متغیر با زمان نیز قابل‌استفاده است، در ادامهٔ این نوشتار، فرض می‌شود که سیستم مورد مطالعه دارای رفتار تغییرناپذیر با زمان بوده و درنتیجه ماتریس‌های F و Q و بردارهای G و H دارای مقادیر ثابتی در طول اجرای الگوریتم هستند. ماتریس F نمایان‌گر مدل تغییر حالت است که به حالت قبلی اعمال می‌شود تا حالت جدید به‌دست آید. بردار G برای محاسبه ماتریس Q به‌منظور استفاده در پیش‌بینی ماتریس کوواریانس خطا مورد استفاده قرار می‌گیرد. در محاسبهٔ ماتریس Q، σ_a^2 واریانس مورد انتظار تغییرات شتاب هدف است که با توجه به ویژگی‌های دینامیک هدف، توسط طراح تعیین شده و مقداری ثابت دارد. بردار Z بردار اندازه‌گیری است که در آن، v_k اغتشاش اندازه‌گیری شامل نوفهٔ گوسی با میانگین صفر و خطای چندی‌سازی است. بردارهای H و R به‌ترتیب نشان‌دهندهٔ ماتریس ضرایب اندازه‌گیری فاصله و سرعت و توان مورد انتظار اغتشاش در فرآیند اندازه‌گیری فاصله و سرعت هستند. مقدار R درعمل توسط طراح و با توجه به دقت مورد انتظار برای اندازه‌گیری‌های مورد استفاده در ورودی تعیین می‌شود. بردار $\hat{X}_0$ شامل تخمین اولیه از مقدار فاصله و سرعت هدف است. ماتریس P_0 نیز معرف مقدار اولیه در نظر گرفته‌شده برای ماتریس کوواریانس حالت است.

۲-۲- مراحل اجرای الگوریتم فیلتر کالمن

چنان‌که درقبل اشاره شد، اجرای الگوریتم شامل دو مرحله پیش‌بینی و به‌هنگام‌سازی است که در ادامه جزئیات هر کدام از این مراحل شرح داده می‌شود.

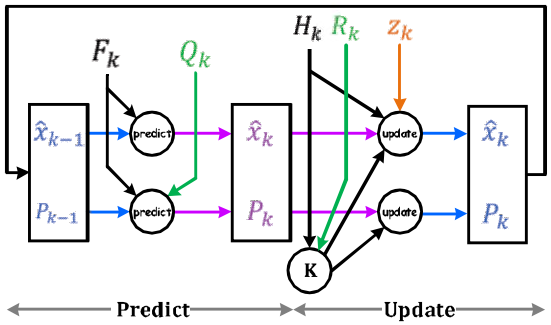
در روابط زیر، ماتریس کوواریانس خطا ماتریسی با تقارن هرمیتی است که آمارگان خطای تخمین را معرفی می‌کند. این ماتریس در هر مرحله از اجرا به شکل بازگشتی محاسبه شده و هیچ‌گونه ارتباطی به مشاهدات ورودی Z_k ندارد. بهرهٔ کالمن نیز مستقل از مشاهدهٔ ورودی در هر مرحله بوده و تعیین‌کنندهٔ میزان تأثیر مشاهدهٔ جدید در تغییر تخمین است. جدول (۲) معادلات فیلتر کالمن در دو مرحلهٔ پیش‌بینی و به‌هنگام‌سازی را نشان می‌دهد.

در روابط این جدول، k اندیس زمان گسسته (شماره مرحله اجرای) فیلتر کالمن است. همچنین $\hat{x}_{k|k-1}$ و $P_{k|k-1}$ به‌ترتیب پیش‌بینی بردار حالت و ماتریس کوواریانس خطا در زمان k با توجه به مشاهدات تا زمان $k-1$ است.

(جدول-۲): معادلات فیلتر کالمن.

(Table-2): Kalman Filter Equations.

مرحله پیش‌بینی	پیش‌بینی بردار حالت	$\hat{x}_{k\|k-1} = F_k\hat{x}_{k-1\|k-1}$
	پیش‌بینی ماتریس کوواریانس خطا	$P_{k\|k-1} = F_k P_{k-1\|k-1}F_k^T + Q_k$
مرحله به‌هنگام‌سازی	محاسبهٔ باقیمانده اندازه‌گیری	$y_k = z_k - H_k\hat{x}_{k\|k-1}$
	محاسبهٔ تغییر کوواریانس	$S_k = H_k P_{k\|k} H_k^T + R_k$
	محاسبهٔ بهرهٔ کالمن بهینه	$K_k = P_{k\|k-1}H_k^T S_k^{-1}$
	تخمین حالت به‌هنگام‌سازی شده	$\hat{x}_{k\|k} = \hat{x}_{k\|k-1} + K_k y_k$
	تخمین کوواریانس به‌هنگام‌سازی شده	$P_{k\|k} = (I - K_k H_k)P_{k\|k-1}$



(شکل-۱): نمودار جعبه‌ای فیلتر کالمن.

(Figure-1): General block diagram of the Kalman filter.

در شکل (۱) نمودار جعبه‌ای فیلتر کالمن، شامل دو مرحلهٔ پیش‌بینی و به‌هنگام‌سازی نشان داده شده است.

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-10]

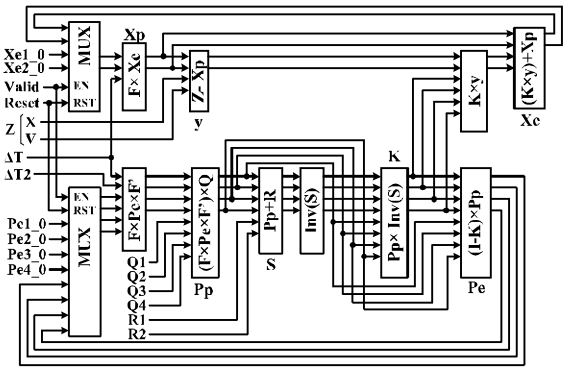
[DOI: 10.29252/jsdp.16.3.100]

همان‌طور که ملاحظه می‌شود، در این ساختار دو مسیر وجود دارد که در یک مسیر تخمین حالت و در مسیر دیگر تخمین کوواریانس به‌هنگام‌سازی می‌شوند.

۳- پیاده‌سازی سخت‌افزاری فیلتر

جهت مدل‌سازی و پیاده‌سازی سخت‌افزاری فیلتر کالمن از نرم‌افزار System Generator شرکت Xilinx استفاده شده است. مدل سخت‌افزاری طراحی‌شده در محیط این نرم‌افزار به‌راحتی به کد سخت‌افزاری قابل پیاده‌سازی روی FPGA تبدیل می‌شود. در شکل (۲) ساختار پیشنهادی برای پیاده‌سازی سخت‌افزاری فیلتر کالمن نشان داده شده است.

به‌منظور افزایش سرعت و کارایی پیاده‌سازی ارائه‌شده در این مقاله، تک‌تک بلوک‌های محاسباتی فیلتر به‌صورت کاملاً سخت‌افزاری طراحی شده و تحقق یافته است. تمامی عملیات مورد نیاز شامل جمع، تفریق و ضرب ماتریسی و نیز محاسبه ماتریس وارون با استفاده از بلوک‌های پایه سخت‌افزاری و DSP48های موجود در FPGA پیاده‌سازی شده است. برای محاسبه ماتریس وارون نیاز به اجرای عملیات تقسیم ممیز ثابت است که این کار با استفاده از تقسیم‌کنندهٔ Radix-2 انجام پذیرفته است.



(شکل-۲): ساختار پیشنهادی برای پیاده‌سازی سخت‌افزاری فیلتر کالمن.

(Figure-2): The proposed architecture for hardware implementation of the Kalman filter.

در طرح ارائه‌شده، از محاسبات ممیز ثابت برای پیاده‌سازی فیلتر کالمن استفاده شده است که در مقایسه با پیاده‌سازی‌های مبتنی بر محاسبات ممیز شناور منجر به افزایش سرعت پردازش و کاهش حجم سخت‌افزار مورد نیاز می‌شود؛ به‌علاوه در طراحی حاضر سعی شده است تا حد ممکن از ساختار خاص مسأله، جهت ساده‌سازی محاسبات و سخت‌افزار مورد نیاز و افزایش سرعت و کارایی نهایی طرح استفاده شود. برای مثال ساختار متقارن و ساده ماتریس H در مسأله مورد بررسی، باعث ساده‌شدن عملیات ضرب ماتریسی می‌شود. در ورودی سیستم از یک تسهیم‌کنندهٔ[1] استفاده شده که در شروع به کار سیستم، مقدار اولیهٔ بردار حالت و مقدار اولیهٔ ماتریس کوواریانس را به ورودی اعمال می‌کند. این مقادیر با "1" بودن پایهٔ بازنشانی[2] و "1" بودن پایهٔ اعتبار[3] ورودی، وارد حلقهٔ الگوریتم می‌شوند.

در این طرح، مقدار "1" پایهٔ بازنشانی به یک ثبّات[4] داده می‌شود و مقدار این ثبّات تا زمانی که پایهٔ اعتبار برای نخستین‌بار "1" شود، همچنان در مقدار "1" ثابت می‌ماند و سپس "0" می‌شود. در تکرارهای بعدی، مقدار پایهٔ بازنشانی "0" است و با "1" بودن پایهٔ اعتبار ورودی، مقادیر محاسبه‌شدهٔ قبلی که درواقع مقادیر به‌هنگام‌سازی‌شدهٔ تخمین بردار $\hat{X}_k$ هستند، برای استفاده در محاسبهٔ بعدی به ورودی تسهیم کننده اعمال می‌شوند. در محاسبات داخلی ساختار پیشنهادی، اعمال سیگنال به دو ورودی تمامی بلوک‌ها شامل ضرب‌کننده‌ها، جمع‌کننده‌ها و تقسیم‌کننده‌ها به‌طور هم‌زمان انجام می‌شود؛ به عبارت دیگر، میزان تأخیر هر دو ورودی یک بلوک با توجه به مقدار تأخیر بلوک‌های محاسباتی موجود در مسیر آن ورودی محاسبه می‌شود. اگر یک ورودی دارای تأخیر کمتر باشد، در مسیر آن تأخیر قرار داده می‌شود تا تأخیر آن با ورودی دیگر برابر شود. مقدار تأخیر کل بلوک پنجاه پالس ساعت است. به عبارت دیگر، پس از پنجاه پالس ساعت، خروجی‌های فیلتر آماده می‌شوند تا در تکرار بعدی مورد استفاده قرار گیرند. به‌همین دلیل، بین دو نمونه متوالی ورودی باید دست‌کم پنجاه پالس ساعت فاصله وجود داشته باشد. با فرض کلاک با فرکانس ۱۲۵ MHz که به‌راحتی بر روی بسیاری از FPGA ها قابل دستیابی است، کل زمانی اجرای فیلتر کالمن برابر ۰/۴ µs به‌دست می‌آید. البته زمان لازم برای هم‌گراشدن خروجی فیلتر کالمن در ابتدای کار، بسیار بیشتر از این مقدار است و به مقادیر انتخاب‌شده برای پارامترهای فیلتر کالمن بستگی دارد.

تعداد بیت‌های تخصیص‌یافته در تمامی مسیر داده ۳۲ بیت، شامل بخش‌های صحیح و اعشاری، است. در هر قسمت سخت‌افزار، با توجه به محدودهٔ تغییرات سیگنال، تعداد بیت تخصیص‌یافته به بخش اعشاری و صحیح متفاوت است.

[1] Multiplexer
[2] Reset
[3] Valid
[4] Register

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-10]
[DOI: 10.29252/jsdp.16.3.100]

مقادیر مورد استفاده برای پارامترهای الگوریتم در جدول (۳) خلاصه شده است. با توجه به این‌که در مسأله مورد بررسی پارامترها دارای مقادیر بسیار کوچکی هستند، جهت کاهش تعداد بیت مورد نیاز اعشار و افزایش دقت محاسبات، مقادیر آن‌ها در یک عدد مناسب بزرگ ضرب می‌شود. در مرحلهٔ محاسبه بهرهٔ کالمن و در هنگام محاسبهٔ وارون ماتریس، مقادیر دوباره به همان عدد تقسیم می‌شوند.

(جدول-۳): مقادیر انتخاب‌شده برای پارامترهای فیلتر کالمن

(Table-3): Selected values for the Kalman filter parameters

متغیر	مقدار
Δt	32 ms
R	$\begin{bmatrix} 1e-12 & 0 \\ 0 & 1e-16 \end{bmatrix}$
σ_a^2	1e-14
$P_0 = \sigma_{x_0}^2 = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_x^2 \end{bmatrix}$	$\begin{bmatrix} 1e-10 & 0 \\ 0 & 1e-10 \end{bmatrix}$

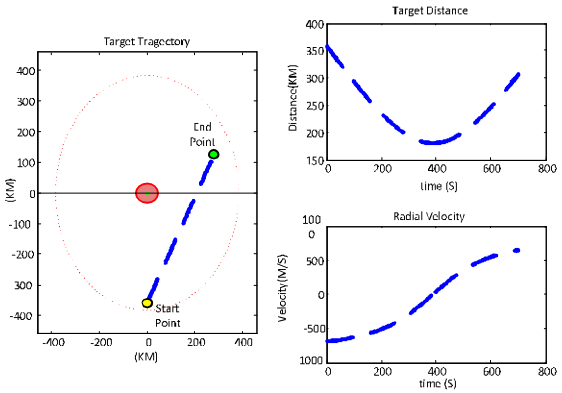
به‌منظور کاهش زمان اجرای فیلتر، بخش‌هایی که به یکدیگر وابستگی داده ندارند، به‌طور هم‌زمان اجرا می‌شوند. به‌عنوان نمونه، بلوک محاسبهٔ بردار y_k در مسیر به‌هنگام‌سازی تخمین حالت و بلوک محاسبهٔ ماتریس کوواریانس در مسیر به‌هنگام‌سازی تخمین کوواریانس مستقل از یکدیگرند و محاسبات آن‌ها به‌صورت هم‌زمان اجرا می‌شود. ماتریس S_k در مسألهٔ مورد بررسی یک ماتریس ۲×۲ است. برای محاسبهٔ وارون ماتریس S_k، باید عکس دترمینان آن محاسبه شود. برای این منظور پس از محاسبه دترمینان ماتریس توسط دو ضرب‌کننده و یک تفریق‌کننده، از یک تقسیم‌کننده برای محاسبهٔ عکس عدد استفاده شده است.

۴- ارزیابی کارایی فیلتر طراحی‌شده

در این بخش نحوهٔ تولید ورودی مناسب برای ارزیابی کارایی فیلتر و نتایج شبیه‌سازی فیلتر سخت‌افزاری ممیز ثابت طراحی‌شده و مقایسه با آن نتایج فیلتر ممیز شناور و نحوهٔ ارزیابی کمّی عملکرد فیلتر با استفاده از معیار NMSE ارائه می‌شود.

۴-۱- تولید ورودی مناسب برای شبیه‌سازی

به‌منظور بررسی کارایی فیلتر طراحی‌شده، یک دنباله از داده‌های متوالی مغشوش‌شده به‌عنوان ورودی نمونه به فیلتر اعمال می‌شود. این ورودی شامل دو مؤلفهٔ فاصله و سرعت شعاعی یک جسم یا هدف متحرک نسبت به محل مشاهده است. سرعت هدف نشان‌دهندهٔ نرخ تغییرات لحظه‌ای فاصله آن از محل مشاهده (مشتق فاصله هدف در راستای شعاعی) است. شکل (۳) تغییرات فاصله و سرعت در داده‌های ورودی را که به‌عنوان نمونه تولید شده‌اند، نشان داده است.



(شکل-۳): مسیر و تغییرات فاصله و سرعت هدف متحرک نسبت به محل مشاهده

(Figure-3): The trajectory of the moving target and its distance and speed variations over time

همان‌طور که ملاحظه می‌شود، هدف در ابتدا به محل مشاهده نزدیک شده و سپس از آن دور می‌شود. متناظر با مسیر تعریف‌شده، سرعت شعاعی هدف در ابتدای حرکت منفی و در انتهای آن مثبت است. پس از تولید مقادیر فاصله و سرعت هدف، خطای چندی سازی و اغتشاش با دامنهٔ زیاد به آن‌ها اضافه می‌شود.

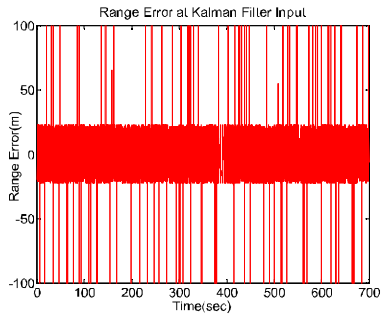
خطای چندی‌سازی افزوده‌شده به سیگنال در این مرحله، درواقع برای مدل‌سازی اندازه‌گیری‌های سرعت و فاصله گسسته‌مقدار است[۱]. در شبیه‌سازی‌های ارائه‌شده در ادامه، گام چندی‌سازی برای اندازه‌گیری فاصله برابر ۴۴ متر (معادل بیشینه خطای چندی‌سازی ۲۲ متر) و برای اندازه‌گیری سرعت برابر ۰/۳ متر بر ثانیه (معادل بیشینه خطای چندی‌سازی ۰/۱۵ متر بر ثانیه) در نظر گرفته شده است. هم‌چنین اغتشاشی برای مدل‌سازی خطای تصادفی به‌نسبه بزرگی که در اندازه‌گیری‌ها سرعت و فاصله می‌تواند به‌وجود آید، اضافه شده است. بیشینه خطای اندازه‌گیری فاصله برابر صد متر و بیشینه خطای اندازه‌گیری سرعت برابر سه متر بر ثانیه لحاظ شده است و هر نمونه از اندازه‌گیری‌های فاصله و سرعت، مستقل از سایر نمونه‌ها، با احتمال ۰/۰۰۵ به این اغتشاش آلوده شده است.

[۱] دقت کنید که در سامانه‌های اندازه‌گیری سرعت و فاصله دیجیتال، خروجی‌های اندازه‌گیری‌شده ذاتاً به‌صورت گسسته‌مقدار در دسترس هستند.

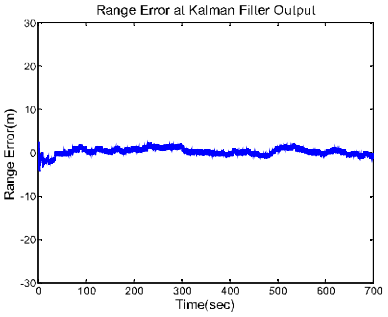
[Downloaded from jsdp.rcisp.ac.ir on 2026-11-10]
[DOI: 10.29252/jsdp.16.3.100]

۲-۴- نتایج شبیه‌سازی

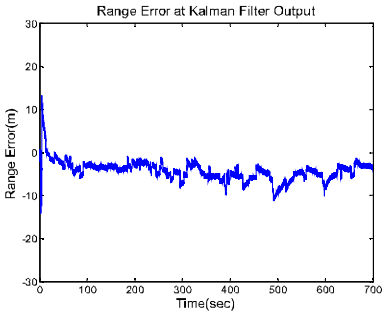
سیگنال‌های نمونه تولیدشده به دو ورودی فاصله و سرعت فیلتر کالمن اعمال شده و نتیجهٔ خروجی سخت‌افزار و خروجی الگوریتم شبیه‌سازی‌شده در نرم‌افزار MATLAB با یکدیگر مقایسه شده است.



(الف)



(ب)



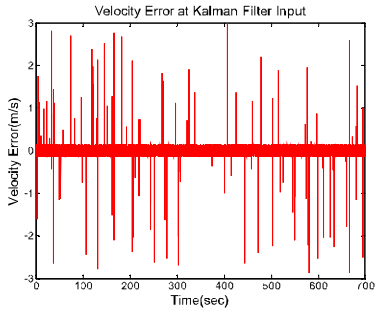
(ج)

(شکل-۴): نمودار خطای تخمین فاصله هدف (الف) بدون فیلتر کالمن (ب) با اعمال فیلتر کالمن در نرم‌افزار (ج) با اعمال فیلتر کالمن در سخت‌افزار.

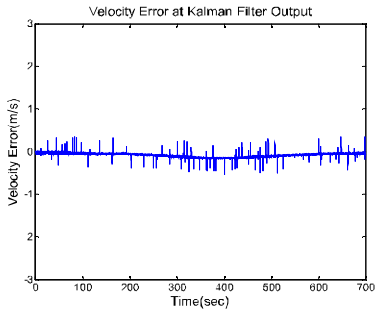
(Figure-4): Diagram of distance estimation error (a) without Kalman filter (b) with the software Kalman filter (c) with the hardware Kalman filter.

شکل (۴-الف) نمودار خطای فاصله اندازه‌گیری‌شده (آغشته به خطای چندی‌سازی و اغتشاش تصادفی) نسبت به فاصله واقعی هدف را نشان می‌دهد. شکل‌های (۴-ب و ج) به‌ترتیب خطای تخمین فاصله هدف در خروجی فیلتر ممیز شناور و خروجی فیلتر کالمن سخت‌افزاری با محاسبات ممیز ثابت، را نشان می‌دهند.

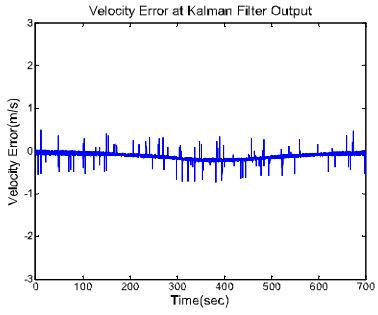
همان‌طور که مشاهده می‌شود، سیگنال اندازه‌گیری‌شده (ورودی فیلتر) دارای خطای چندی‌سازی با دامنه‌ای در حدود ۲۲ متر است. هم‌چنین به‌دلیل اغتشاش تصادفی اضافه‌شده به سیگنال، شکل آن دارای برخی تغییرات ضربه‌ای نیز هست. در خروجی متناظر با تخمین فاصله در هر دو نوع پیاده‌سازی فیلتر کالمن (شکل‌های (۴-ب و ج))، خطای ناشی از چندی‌سازی تا حد زیادی کاهش یافته و درواقع فیلتر کالمن به‌طور مؤثری باعث هموارسازی تخمین فاصله شده است. البته با توجه به شکل دیده می‌شود که به واسطه خطای محاسبات ممیز ثابت در خروجی فیلتر کالمن سخت‌افزاری، میزان خطای تخمین فاصله در این فیلتر کمی بیشتر از فیلتر ممیز شناور است.



(الف)



(ب)



(ج)

(شکل-۵). نمودار خطای تخمین سرعت هدف (الف) بدون فیلتر کالمن (ب) با اعمال فیلتر کالمن در نرم‌افزار (ج) با اعمال فیلتر کالمن در سخت‌افزار.

(Figure-5): Diagrams of velocity estimation error (a) without Kalman filter (b) with the software Kalman filter (c) with the hardware Kalman filter.

شکل (۵-الف) نمودار خطای سرعت اندازه‌گیری شده (آغشته به خطای چندی سازی و اغتشاش تصادفی) نسبت به سرعت واقعی هدف را نشان می‌دهد. شکل‌های (۵-ب و ج)

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-10]

[DOI: 10.29252/jsdp.16.3.100]

به‌ترتیب خطای تخمین سرعت هدف در خروجی فیلتر ممیز شناور و خروجی فیلتر کالمن سخت‌افزاری با محاسبات ممیز ثابت، را نشان می‌دهد.

همان‌طور که مشاهده می‌شود، اعمال فیلتر کالمن تا حد زیادی اثر اغتشاش تصادفی در آشکارسازی سرعت را از بین برده و اثر خطای چندی‌سازی را نیز تا حد بسیار زیادی کاهش داده است. همچنین می‌توان دید که نتایج فیلتر کالمن سخت‌افزاری به‌طورتقریبی مشابه فیلتر کالمن ممیز شناور طراحی‌شده در متلب است. تفاوت اندک خطا در شکل‌های (۵- ب و ج) به‌دلیل خطای محاسبات ممیز ثابت است.

۳-۴- مقایسه کمّی نتایج

در این بخش از پارامترهای میانگین نرمالیزه شده مربعات خطا (NMSE[1]) و بیشینه خطا برای بررسی کمّی عملکرد فیلتر کالمن و مقایسه پیاده‌سازی سخت‌افزاری ممیز ثابت با پیاده‌سازی نرم‌افزاری ممیز شناور استفاده می‌شود.

پارامتر NMSE با استفاده از رابطه (۱) محاسبه می‌شود که در آن X سیگنال ورودی قبل از آغشته‌شدن به اغتشاش (مقدار واقعی کمیت‌های فاصله و سرعت) و $\hat{X}$ تخمین ارائه‌شده در خروجی فیلتر کالمن از این کمیت‌ها است.

$$\mathrm{NMSE} = \frac{\|X - \hat{X}\|_2^2}{\|X\|_2^2} \qquad (۱)$$

نتایج شبیه‌سازی نشان می‌دهد که وقتی از فیلتر کالمن برای کاهش خطا و هموارسازی داده‌های آشکارشده استفاده نشود، مقدار NMSE برای خروجی فاصله برابر ۳۳/۴ dB- و برای خروجی سرعت برابر ۳۵ dB- است. مقدار NMSE بعد از اعمال فیلتر کالمن نرم‌افزاری ممیز شناور برای خروجی فاصله برابر ۴۹/۳ dB- و برای خروجی سرعت ۳۷ dB- است که به‌طور میانگین (با معیار NMSE) بهبودی در حدود ۱۶ dB برای فاصله و حدود ۲ dB برای سرعت نشان می‌دهد. همچنین اعمال فیلتر کالمن باعث می‌شود تا بیشینه خطای تخمین فاصله از حدود صد متر (در اندازه‌گیری اولیه) به کمتر از چهار متر (در خروجی فیلتر) و بیشینه خطای تخمین سرعت از حدود سه متر بر ثانیه به کمتر از ۰/۵ متر بر ثانیه کاهش یابد.

مقدار بهبود در تخمین فاصله نسبت به تخمین سرعت چشم‌گیرتر است، زیرا به‌دلیل رابطه مشتق-انتگرالی سرعت و فاصله، تأثیر داده‌های آشکارسازی‌شده سرعت در روابط تخمین فاصله قابل‌توجه‌تر است.

مقدار NMSE بعد از اعمال فیلتر کالمن سخت‌افزاری ممیز ثابت برای خروجی فاصله برابر ۴۴/۳ dB- (معادل بهبود میانگین در حدود ۱۱ dB) و برای خروجی سرعت برابر ۳۶/۲ dB- (معادل بهبود میانگین در حدود ۱/۲ dB) است. هم‌چنین اعمال فیلتر کالمن سخت‌افزاری باعث می‌شود تا بیشینه خطای تخمین فاصله از حدود صد متر (در اندازه‌گیری اولیه) به کمتر از ده متر (در خروجی فیلتر) و بیشینه خطای تخمین سرعت از حدود سه متر بر ثانیه به کمتر از ۰/۷ متر بر ثانیه کاهش یابد.

مقایسه پارامترهای کمّی فیلتر کالمن سخت‌افزاری ممیز ثابت با فیلتر ممیز شناور نشان می‌دهد که پیاده‌سازی سخت‌افزاری عملکردی قابل‌قبول و بسیار نزدیک به نمونهٔ نرم‌افزاری دارد. توجه کنید که رسیدن به این نتیجه با انتخاب مناسب طول بیت قسمت صحیح و اعشاری در محاسبات داخلی فیلتر امکان‌پذیر شده است.

۴-۴- نتایج پیاده‌سازی بر روی برد FPGA

در بخش قبل، نتایج حاصل از شبیه‌سازی فیلتر سخت‌افزاری ممیز ثابت طراحی‌شده ارائه شد. در این بخش مقدار منابع مصرفی برای پیاده‌سازی فیلتر کالمن پس از سنتز سخت‌افزار بر روی برد ZC706 EVB (XC7z045ffg900-2) در جدول (۴) خلاصه شده است:

(جدول-۴): منابع مصرفی فیلتر کالمن پس از سنتز سخت‌افزار بر روی برد ارزیابی ZC706.

(Table-4): Hardware resources utilization of the Kalman filter after synthesize on the evaluation board ZC706.

Resource	Estimation	Available	% Utilization
FF	7416	437200	1.70
LUT	9265	218600	4.24
Memory LUT	708	70400	1.01
DSP48	120	900	13.33
BUFG	1	32	3.13

میزان مصرف توان دینامیک سخت‌افزار طراحی‌شده با پالس ساعت ۱۲۵ MHz برابر ۴۶۸ mW تخمین زده شده است. شکل (۶) نمایی از برد مورد استفاده برای پیاده‌سازی این سیستم را نشان می‌دهد.

[1] Normalized Mean Square Error

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-10]

[DOI: 10.29252/jsdp.16.3.100]



(شکل-۶): نمایی از برد Zynq-Z706 مورد استفاده برای پیاده‌سازی سخت‌افزاری فیلتر کالمن.

(Figure-6): A view of the Zynq-Z706 board used to implement the Kalman filter.

۵-۴- مقایسه با برخی کارهای انجام‌شده

روش ارائه‌شده در [21]، مشابه روند ارائه‌شده در این مقاله، از روش‌های متداول بهینه‌سازی پیاده‌سازی سخت‌افزار مانند راه‌کارهای جریان داده، خط لوله و گشودن حلقه‌ها استفاده شده تا پیاده‌سازی مؤثری از فیلتر کالمن گسترش‌یافته (EKF) بر روی FPGA محقق شود. در این مقاله، اگرچه پژوهش‌گران توانسته‌اند با حجم قابل قبولی از منابع سخت‌افزاری (که قابل مقایسه با منابع مصرف‌شده در پیاده‌سازی حاضر است) فیلتر کالمن گسترش‌یافته را -که پیچیدگی اندکی بیشتر در مقایسه با فیلتر کالمن معمولی دارد- پیاده‌سازی کنند؛ اما در کمترکردن زمان اجرای الگوریتم از حد یک میلی ثانیه ناکام بوده‌اند. این زمان در مقایسه با زمان اجرای الگوریتم در روش ما، که به‌کمک رویکرد توزیع‌شده و تمهیدات اتخاذشده در پیاده‌سازی به‌شدت و تا حد چند میکروثانیه کاهش یافته، بسیار بزرگ به نظر می‌رسد، هم‌چنین از نظر نحوه پیاده‌سازی، رویکرد پی‌گرفته‌شده در مقاله حاضر کوشیده است تا حد امکان بهینه‌سازی سخت‌افزاری و تعیین طول بیت مناسب برای قسمت‌های مختلف در پایین‌ترین سطح ممکن و در حین فرآیند طراحی سخت‌افزار انجام شود. در مقابل، در [21] از ابزارهای سنتز سطح بالا مانند Vivado HLS برای پیاده‌سازی طرح استفاده شده است که در حالت کلی میزان بهینه‌سازی محدودتری را فراهم می‌کند.

در [22]، روشی برای بهینه‌سازی پیاده‌سازی فیلتر کالمن شبیه‌سازی‌شده معرفی شده و نشان داده شده که سرعت عملکرد مناسب با استفاده از حجم محدودی از منابع سخت‌افزاری قابل تحقق است. با این وجود، مسأله درنظرگرفته‌شده برای ارزیابی کارایی در این مقاله، یک مسأله یک‌بعدی و بسیار ساده بوده و معیار مناسبی برای مقایسه در اختیار نمی‌گذارد؛ به‌علاوه در پیاده‌سازی پیشنهادی در این مقاله کاربرد Block RAM های اضافی ضروری است.

روش ارائه‌شده در [23]، مسأله‌ای مشابه مسأله مطرح‌شده را در مقاله حاضر پوشش می‌دهد. این پژوهش از فیلتر کالمن گسترش‌یافته برای حل مسأله ردگیری و تعقیب هدف بهره برده و به زمان اجرای کمتر از دوازده میکروثانیه برای هر اجرای فیلتر دست می‌یابد. اگرچه این زمان بزرگ‌تر از زمان اجرای به‌دست‌آمده در پیاده‌سازی ما است، اما با توجه به استفاده از فیلتر کالمن گسترش‌یافته و ابعاد بزرگ‌تر بردار حالت مورد استفاده، نتیجه جالبی به نظر می‌رسد؛ به‌علاوه شایان توجه است که طراحی ارائه‌شده در [23] از بلوک‌های اختصاصی پردازش ممیز شناور در FPGA استفاده کرده است که باعث می‌شود منابع سخت‌افزاری و هزینه پیاده‌سازی افزایش یابد. در طرح ارائه‌شده در این مقاله، با حفظ دقت محاسباتی مورد نیاز، فقط از بلوک‌های ممیز ثابت استفاده شده که موجب مصرف منابع و هزینه کمتری است.

در [24]، دو پیاده‌سازی مبتنی بر DSP و FPGA از نسخه‌های مناسب فیلتر کالمن برای تعقیب اهداف مانورکننده ارائه شده است. در این مقاله نشان داده شده است که پیاده‌سازی بر روی FPGA، مطابق انتظار، بسیار سریع‌تر و مؤثرتر بوده و به زمان اجرای ۳۲ میکروثانیه برای هر تکرار فیلتر (در مقایسه با زمان بیش از دو میلی ثانیه بر روی DSP) دست می‌یابد که البته همچنان در مقایسه با عدد به‌دست‌آمده در پیاده‌سازی ما بزرگ‌تر است.

۵- نتیجه‌گیری

در این مقاله، طراحی مناسب و پیاده‌سازی سخت‌افزاری یک فیلتر کالمن با محاسبات ممیز ثابت به‌منظور کاهش خطای سیگنال‌های دریافتی فاصله و سرعت شعاعی یک هدف متحرک مورد بررسی قرار گرفت. در روند پیاده‌سازی سخت‌افزاری سعی شده است تا با موازی‌سازی قسمت‌هایی که وابستگی داده ندارند، زمان لازم برای انجام محاسبات فیلتر تا حد امکان کاهش یابد. در نتیجه این تمهید، در پیاده‌سازی فیلتر زمان لازم برای انجام محاسبات هر مرحله اجرایی برابر ۰/۴ µs (معادل پنجاه پالس ساعت با فرکانس ۱۲۵ MHz) به‌دست آمده است. بررسی عملکرد فیلتر با استفاده از پارامتر NMSE نشان می‌دهد دقت پیاده‌سازی فیلتر کالمن به‌صورت سخت‌افزاری ممیز ثابت بسیار نزدیک به نمونه نرم‌افزاری ممیز شناور آن است و تا حد زیادی موجب کاهش خطای تخمین فاصله و سرعت هدف می‌شود.

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-10]

[DOI: 10.29252/jsdp.16.3.100]

6- References ۶- مراجع

[1] R. Kalman, "A new approach to linear filtering and prediction problems," Trans. ASME, J. Basic Eng., vol. 82, pp. 35–45, 1960, series D.

[۲] حنیفه‌لو زهرا، منجمی سید امیرحسن، معلم پیمان، "ارائهٔ روشی مقاوم نسبت به تغییرات روشنایی در آشکارسازی و ردیابی خودروها درصحنه‌های ترافیکی،" مجله پردازش علائم و داده‌ها. ۱۳۹۵; ۱۳ (۳) :۷۹-۹۸.

[2] Z. Hanifelou, S. A. H. Monadjemi and P. Moallem, "Robust method of changes of light to detect and track vehicles in traffic scenes" *Journal of Signal and Data Processing (JSDP)*, vol. 13, no. 3, pp. 79-98, 2016.

[۳] شایق فرزانه، قاسمی فهیمه، امیر فتاحی رسول، صدری سعید، انصاری اصل کریم، "پیش‌گویی برخط و تک‌کاناله وقوع حمله‌های صرعی با ارائه الگوی تولید صرع بر روی سیگنال‌های depth-EEG با استفاده از فیلتر کالمن توسعه‌یافته،" مجله پردازش علائم و داده‌ها. ۱۳۹۷; ۱۵ (۱) :۳-۲۸.

[3] F. Shayegh, F. Ghasemi, R. Amirfattahi, S. Sadri ,and K. Ansariasl, "Online Single-Channel Seizure Prediction, Based on Seizure Genesis Model of Depth-EEG Signals Using Extended Kalman Filter," *Journal of Signal and Data Processing (JSDP)*, vol. 15, no. 1, pp. 3-28, 2018.

[4] M. Grewal and A. Andrews, *Kalman Theory, Theory and Practice Using MATLAB.* 3rd ed. Hoboken, NJ: Wiley, 2008.

[5] M. Verhaegen and P. V. Dooren, "Numerical aspects of different Kalman filter implementations," *IEEE Trans. Autom. Control*, vol. AC 31, no. 10, pp. 907–917, Oct. 1986.

[6] V. Smidl and Z. Peroutka, "Advantages of square-root extended Kalman filter for sensorless control of AC drives," *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4189–4196, Nov. 2012.

[7] J. Mendel, "Computational requirements for a discrete Kalman filter," *IEEE Trans. Autom. Control*, vol. AC-16, no. 6, pp. 748–758, Dec. 1971.

[8] M. Hilairet, F. Auger, and C. Darengosse, "Two efficient Kalman filters for flux and velocity estimation of induction motors," in *Proc. IEEE Power Electron. Spec. Conf.*, Jun. 2000, vol. 2, pp. 891–896.

[9] L. Idkhajine and E. Monmasson, "Design methodology for complex FPGA-based controllers—Application to an EKF sensorless ac drive," in *Proc. XIX ICEM*, Sep. 2010, pp. 1–6.

[10] J. Keller and M. Darouach, "Two-stage Kalman estimator with unknown exogenous inputs," *Automatica*, vol. 35, no. 2, pp. 339–342, Feb. 1999.

[11] C. Hsieh and F. Chen, "Optimal solution of the two-stage Kalman estimator," *IEEE Trans. Autom. Control*, vol. 44, no. 1, pp. 194–199, Jan. 1999.

[12] C. Hsieh, "General two-stage extended Kalman filters," *IEEE Trans. Autom. Control*, vol. 48, no. 2, pp. 289–293, Feb. 2003.

[13] M. Hilairet, F. Auger, and E. Berthelot, "Speed and rotor flux estimation of induction machines using a two-stage extended Kalman filter," *Automatica*, vol. 45, no. 8, pp. 1819–1827, Aug. 2009.

[14] A. Akrad, M. Hilairet, and D. Diallo, "Design of a fault-tolerant controller based on observers for a PMSM drive," *IEEE Trans. Ind. Electron.*, vol. 58, no. 4, pp. 1416–1427, Apr. 2011.

[15] S. Bolognani, R. Oboe, and M. Zigliotto, "Sensorless full-digital PMSM drive with EKF estimation of speed and rotor position," *IEEE Trans. Ind. Electron.*, vol. 46, no. 1, pp. 184–191, Feb. 1999.

[16] H.-G. Yeh, "Systolic implementation on Kalman filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, no. 9, pp. 1514–1517, Sep. 1988.

[17] S.-Y. Kung and J.-N. Hwang, "Systolic array designs for Kalman filtering," *IEEE Trans. Signal Process.*, vol. 39, no. 1, pp. 171 182, Jan. 1991.

[18] E. Monmasson, L. Idkhajine, M. Cirstea, I. Bahri, A. Tisan, and M. Naouar, "FPGAs in industrial control applications," *IEEE Trans. Ind. Informat.*, vol. 7, no. 2, pp. 224–243, May 2011.

[19] C. Lee and Z. Salcic, "High-performance FPGA-based implementation of Kalman filter," *Microprocess. Microsyst.*, vol. 21, no. 4, pp. 257 265, Dec. 1997.

[20] L. Idkhajine, E. Monmasson, and A. Maalouf, "Fully FPGA-based sensorless control for synchronous AC drive using an extended Kalman filter," *IEEE Trans. Ind. Electron.*, vol. 59, no. 10, pp. 3908–3918, Oct. 2012.

[21] A. Jarrah, A. Al-Tamimi, and T. Albashir, "Optimized parallel implementation of Extended Kalman filter using FPGA," *Journal of Circuits, Systems and Computers*, vol. 01, no. 27 no. 01, June 2017.

[22] N. Noordin, Z. Ibrahim, M. H. J. Xie, R. Samad and N. Hasan, "FPGA implementation of simulated Kalman filter optimization algo-

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-10]

[DOI: 10.29252/jsdp.16.3.100]

rithm," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, VOL. 10, no. 1-3, pp. 21-24, 2018.

[23] D. Pritsker, "Hybrid implementation of Extended Kalman Filter on an FPGA," in *Proc. IEEE Radar Conf. (RadarCon)*, pp. 0077-0082, 2015.

[24] P. L. Wu, L. Z. Zhang and X. Y. Zhang, "The design of DSP/FPGA based maneuvering target tracking system," *WSEAS Trans. Circ. Syst.*, vol. 13, pp. 75-84, 2014.



شهاب‌الدین رحمانیان مدرک کارشناسی خود را در سال ۱۳۸۳ از دانشگاه صنعتی اصفهان، مدرک کارشناسی ارشد خود را در سال ۱۳۸۶ از دانشگاه تهران و مدرک دکترای خود را در سال ۱۳۹۲ از دانشگاه تربیت مدرس، هر سه در رشته مهندسی برق-الکترونیک دریافت کرده است. ایشان هم‌اکنون استادیار پژوهشکده اویونیک دانشگاه صنعتی اصفهان است. زمینه‌های پژوهشی مورد علاقه ایشان شامل طراحی و پیاده‌سازی سامانه‌های ردگیری هدف و پیاده‌سازی سامانه‌های پردازش سیگنال‌های دیجیتال بر روی FPGA است.

نشانی رایانامه ایشان عبارت است از:

Rahmanian@cc.iut.ac.ir



محمدحسین باطنی مدرک کارشناسی، کارشناسی ارشد و دکترای خود را در مهندسی برق به‌ترتیب در سال‌های ۱۳۸۴، ۱۳۸۶ و ۱۳۹۷ از دانشگاه صنعتی اصفهان دریافت کرد. برخی از زمینه‌های پژوهشی موردعلاقه وی عبارت‌اند از: سامانه‌های مخابراتی، مخابرات بی‌سیم، پردازش سیگنال‌های مخابراتی، شبکه‌های مخابراتی و الگوریتم‌های توزیع‌شده و پردازش دیجیتالی سیگنال‌ها.

نشانی رایانامه ایشان عبارت است از:

mh.batcni@cc.iut.ac.ir



مجدالدین نجفی مدارک کارشناسی، کارشناسی ارشد و دکترای خود را در زمینه مهندسی برق-کنترل از دانشگاه صنعتی اصفهان و به‌ترتیب در سال‌های ۱۳۸۴، ۱۳۸۶ و ۱۳۹۱ اخذ کرد. از سال ۱۳۹۳، وی استادیار پژوهشکده اویونیک دانشگاه صنعتی اصفهان است. زمینه‌های پژوهشی موردعلاقه وی عبارت‌اند از: سامانه‌های چندعاملی، سامانه‌های تأخیر زمانی، کنترل مقاوم و ناوبری و ردیابی چند هدفی.

نشانی رایانامه ایشان عبارت است از:

majd_najafi@cc.iut.ac.ir



محمد فرداد مدرک کارشناسی، کارشناسی ارشد و دکترای خود را در رشته مهندسی برق گرایش الکترونیک به‌ترتیب در سال‌های ۱۳۸۵، ۱۳۸۹ و ۱۳۹۷ از دانشگاه گیلان، دانشگاه تبریز و دانشگاه صنعتی اصفهان دریافت کرده است. زمینه‌های پژوهشی مورد علاقه وی طراحی و پیاده‌سازی سامانه‌های پردازش سیگنال، پیاده‌سازی سامانه‌های دیجیتال بر روی FPGA و پردازش سخت‌افزاری تصاویر دیجیتال است.

نشانی رایانامه ایشان عبارت است از:

m.fardad@ec.iut.ac.ir

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-10]

[DOI: 10.29252/jsdp.16.3.100]