

(Research Article)

Implementation of the direction of arrival estimation algorithms by means of GPU-parallel processing in the Kuda environment

A. Ghasemi^{*1}, H. Sadeghi², M. Shahrab³

1. Department of Computer and Electrical Engineering, Shahid Beheshti University

2. Infrasound Research Laboratory

3. Department of Electrical and Computer Engineering, Isfahan University of Technology

Received: 2022/05/16, Accepted: 2022/07/11

DOR 20.1001.1.23455748.1401.10.1.3.2

Abstract

Direction-of-arrival (DOA) estimation of audio signals is critical in different areas, including electronic war, sonar, etc. The beamforming methods like Minimum Variance Distortionless Response (MVDR), Delay-and-Sum (DAS), and subspace-based Multiple Signal Classification (MUSIC) are the most known DOA estimation techniques. The mentioned methods have high computational complexity. Hence using the algorithms with high computational complexity in the real-time DOA estimation applications is a serious challenge. On the other hand, the main characteristic of the methods is their high potential for parallelization. The objective of this paper is a parallel implementation of the considered algorithms using a Graphics Processing Unit (GPU) instead of a Central Processing Unit (CPU) for increasing execution speed and real-time implementation of the mentioned algorithms. To this aim, the Kuda programming model is used to implement the algorithm on a GPU. This algorithm is also implemented serially in MATLAB to investigate the parallel implementation performance. The results show that parallel implementation of these algorithms can increase the program execution time ten times more than serial implementation. Accuracies of different implementations are validated using simulations by MATLAB and Kuda.

Keywords: Signal Orientation, Beamforming, Parallel Processing, Graphics Processing Unit (GPU), Kuda Programming Model.

pp. 23-33 (In Persian)

* Corresponding author E-mail: intelligentcontrol.ghasemi@gmail.com

پیاده‌سازی الگوریتم‌های جهت‌یابی به صورت پردازش موازی به کمک واحدهای پردازنده ترسیمی (جی‌پی‌یو) در محیط کودا

امیر قاسمی^{۱*}، حامد صادقی^۲، مجتبی شهراب^۳

۱. دانشکده مهندسی برق و کامپیوتر، دانشگاه شهید بهشتی

۲. آزمایشگاه تحقیقاتی فروصدا

۳. دانشکده مهندسی برق و کامپیوتر، دانشگاه صنعتی اصفهان

دریافت: ۱۴۰۱/۰۲/۲۶، پذیرش: ۱۴۰۱/۰۴/۲۰

چکیده

جهت‌یابی علامت صوتی در حوزه‌های مختلفی از قبیل جنگ الکترونیک، سونار و غیره از اهمیت بسیار بالایی برخوردار است. روش‌های شکل‌دهی پرتو مانند ام‌وی‌دی‌آر و دی‌ای‌اس و روش مبتنی بر زیرفضای موزیک، از شناخته‌شده‌ترین روش‌های جهت‌یابی علامت به‌شمار می‌آیند. پیچیدگی محاسباتی بالای روش‌های ذکر شده باعث می‌شود که استفاده از این الگوریتم‌ها در کاربردهایی که جهت‌یابی زمان حقیقی نیاز است، با چالش جدی مواجه شود. از سوی دیگر، یک ویژگی مهم روش‌های فوق، پتانسیل بالای آن‌ها برای موازی‌سازی است. هدف این مقاله، پیاده‌سازی موازی الگوریتم‌های نام‌برده با به‌کارگیری واحد پردازنده گرافیکی (جی‌پی‌یو) به جای واحد پردازنده مرکزی (سی‌پی‌یو)، به منظور افزایش سرعت اجرا و رسیدن به حالت زمان حقیقی می‌باشد. برای دست‌یابی به این هدف، از الگوی برنامه‌نویسی کودا برای پیاده‌سازی الگوریتم روی پردازنده گرافیکی استفاده شده است. به منظور بررسی عملکرد پیاده‌سازی موازی، این الگوریتم‌ها در نرم‌افزار متلب نیز به صورت سریال پیاده‌سازی شدند. نتایج به‌دست‌آمده نشان می‌دهند که پیاده‌سازی موازی این الگوریتم‌ها می‌تواند تا بیش از ده برابر سرعت اجرای برنامه را نسبت به حالت سریال افزایش دهد. صحت عملکرد پیاده‌سازی‌های مختلف، در هر دو محیط متلب و کودا توسط داده‌های شبیه‌سازی‌شده تایید گردید.

کلیدواژه‌ها: جهت‌یابی علامت، شکل‌دهی پرتو، پردازش موازی، واحد پردازنده گرافیکی (جی‌پی‌یو)، الگوی برنامه‌نویسی کودا.

۱. مقدمه

منتشرشده از هدف و دریافت آن توسط آرایه‌ای از

حسگرها انجام می‌گیرد.

تاکنون روش‌های مختلفی برای جهت‌یابی پیشنهاد شده است که از مهم‌ترین آن‌ها می‌توان به روش تاخیر و جمع (دی‌ای‌اس)^۱ [۲]، الگوریتم کمینه واریانس بدون اعوجاج (ام‌وی‌دی‌آر)^۲ [۲]، الگوریتم طبقه‌بندی علامت چندگانه (موزیک)^۳ [۳] و روش‌های مبتنی بر نمایش تنک علامت [۴-۵] اشاره کرد. در این روش‌ها، جهت‌یابی با استفاده از داده‌های زیاد به این منظور صورت می‌گیرد که باعث بالا رفتن حجم محاسبات می‌شود. علاوه بر این، به‌منظور

جهت‌یابی علامت‌های صوتی یکی از موضوعاتی است که امروزه ذهن بسیاری از محققان را به خود مشغول کرده است. در جهت‌یابی صوتی، هدف رهگیری جسمی است که منبع علامت صوتی بوده و یا امواج صوتی ارسالی را منعکس می‌نماید. روش‌های جهت‌یابی صوتی، اغلب مبتنی بر آرایه فازی و شکل‌دهی پرتو هستند [۱]. در این نوع جهت‌یابی، از امواج صوتی دریافتی حسگرهای آرایه به‌منظور تعیین جهت دقیق منبع استفاده می‌شود. به این شکل که در آن، جهت‌یابی براساس تغییرات علامت

¹ DAS

² MVDR

³ MUSIC

* نویسنده پاسخگو: intelligentcontrol.ghasemi@gmail.com

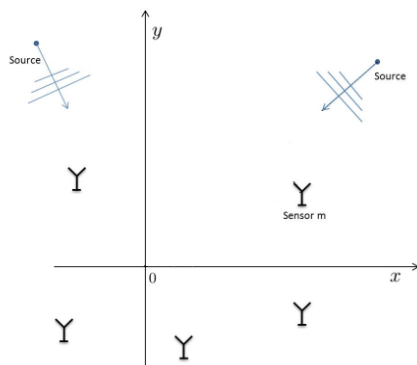
بردار یا یک متغیر تصادفی می‌باشد. علائم $(\cdot)^T$ ، $(\cdot)^*$ و $(\cdot)^H$ به ترتیب نشان‌دهنده ترانهاده، مزدوج و مزدوج ترانهاده یک بردار یا یک ماتریس می‌باشند. عملگر $\text{trace}\{\cdot\}$ درایه‌های روی قطر اصلی یک ماتریس را جمع می‌کند. I_M یک ماتریس یکه با ابعاد $M \times M$ بوده و $\|\cdot\|_F$ نرم دوم یک بردار را نشان می‌دهد.

۲. مواد و روش‌ها

۲-۱. الگوی علامت

مطابق شکل ۱، یک آرایه با ساختار دلخواه، شامل M حسگر در نظر گرفته می‌شود. مکان قرارگیری حسگر m -ام با $(P_{x_m}, P_{y_m}, P_{z_m})$ مشخص می‌شود. اگر K علامت باند باریک از زاویه‌های افقی $\{\theta_1, \dots, \theta_K\}$ و زاویه‌های عمودی $\{\phi_1, \dots, \phi_K\}$ به این آرایه تابیده شود، علامت دریافتی آرایه به‌صورت زیر نشان داده خواهد شد:

$$x(t) = A(\theta, \phi)s(t) + v(t) \quad (1)$$



شکل ۱ یک آرایه شامل M حسگر [۶].

در رابطه (۱)، $A_{M \times K}(\theta, \phi)$ ماتریس هادی آرایه، $s_{K \times 1}(t)$ بردار علامت منابع و $v_{M \times 1}(t)$ بردار نوفه آنتن‌ها می‌باشند. نوفه آنتن‌ها دارای توزیع گوسی با میانگین صفر و ماتریس کواریانس $\sigma^2 I_M$ در نظر گرفته می‌شود. اگر مبدا مختصات به‌عنوان مرجع تاخیر زمانی در نظر گرفته شود، تاخیر انتشار علامت k -ام در حسگر m -ام به‌صورت رابطه زیر تعریف می‌شود:

$$\tau_{ksm} \triangleq \frac{P_{x_m} \sin \phi_k \cos \theta_k}{c} + \frac{P_{y_m} \sin \phi_k \sin \theta_k}{c} + \frac{P_{z_m} \cos \phi_k}{c} \quad (2)$$

تعیین جهت، نیاز به جستجو در تمام زاویه‌ها می‌باشد که بسیار زمان‌بر است. حجم زیاد محاسبات لازم برای پردازش علامت‌های دریافتی، معمولاً مانع از اجرای زمان حقیقی الگوریتم‌های موجود می‌شود. راهکار برطرف کردن این مشکل، افزایش سرعت پردازش علامت با استفاده از اجرای موازی الگوریتم‌های جهت‌یابی، توسط چندین واحد پردازش‌گر است.

واحد پردازنده مرکزی رایانه (سی‌پی‌یو)^۱، همه‌ی الگوریتم‌ها را به‌صورت خط به خط و پشت سر هم اجرا می‌کند؛ بنابراین قابلیت اجرای موازی در سی‌پی‌یو بسیار محدود است. این در حالی است که جی‌پی‌یو یا همان کارت گرافیکی رایانه قابلیت زیادی در اجرای هم‌زمان و موازی دارد. بنابراین هدف اصلی استفاده از جی‌پی‌یو به جای سی‌پی‌یو، افزایش سرعت و اجرای زمان حقیقی الگوریتم‌های جهت‌یابی می‌باشد. در راستای این هدف، الگوریتم‌های دی‌ای‌اس، ام‌وی‌دی‌آر و موزیک روی واحد پردازنده‌ی گرافیکی پیاده‌سازی شده و نتایج حاصل از آن با یکدیگر و هم‌چنین با نتایج حاصل از اجرای آن‌ها روی واحد پردازنده‌ی مرکزی مقایسه می‌شوند. بنابراین، نوآوری‌های این مقاله به شرح زیر است:

- ۱) پیاده‌سازی موازی الگوریتم‌های شکل‌دهی پرتو مانند ام‌وی‌دی‌آر و دی‌ای‌اس و روش مبتنی بر زیر فضای موزیک با به کارگیری واحد پردازنده گرافیکی (جی‌پی‌یو) به‌جای واحد پردازنده مرکزی (سی‌پی‌یو)،
- ۲) افزایش سرعت پردازش و رسیدن به حالت اجرای زمان حقیقی الگوریتم‌های شکل‌دهی پرتو،
- ۳) مقایسه سرعت عملکرد الگوریتم‌های جهت‌یابی به‌هنگام پیاده‌سازی موازی و سریال در دو محیط برنامه‌نویسی متلب و کودا.

۲-۱. نشانه‌ها

در این پاراگراف، نشانه‌های استفاده‌شده در مقاله معرفی می‌شوند. حروف کوچک ضخیم برای نمایش بردارها و حروف بزرگ ضخیم برای نمایش ماتریس‌ها استفاده خواهند شد. علامت $E\{\cdot\}$ نشان‌دهنده میانگین آماری یک

¹ CPU

در الگوریتم موزیک از تجزیه ویژه ماتریس همبستگی برای تخمین زاویه ورود استفاده می‌شود. تجزیه ویژه ماتریس کواریانس علامت دریافتی را می‌توان به صورت زیر محاسبه نمود:

$$R_x = U_s \Lambda_s U_s^H + \sigma^2 U_n U_n^H \quad (5)$$

در رابطه فوق، U_n و U_s به ترتیب بردار ویژه‌های تشکیل‌دهنده زیرفضای علامت و نوفه هستند. Λ_s یک ماتریس قطری که درایه‌های روی قطر اصلی آن توان منابع و σ^2 واریانس نوفه می‌باشند.

در الگوریتم موزیک از تعامد زیرفضای علامت و نوفه برای جهت‌یابی استفاده می‌شود. مقادیر ویژه ماتریس R_x را می‌توان به صورت زیر نمایش داد.

$$\{p_1 + \sigma^2, p_2 + \sigma^2, \dots, p_K + \sigma^2, \sigma^2, \dots, \sigma^2\}$$

مشاهده می‌شود که $M-K$ مورد از مقادیر ویژه‌ها برابر σ^2 هستند. می‌توان با استفاده از بردارهای ویژه نظیر مقادیر ویژه‌های کوچک ماتریس R_x ، زیرفضای نوفه را تعیین کرد. با توجه به اینکه زیرفضای علامت با استفاده از بردارهای هادی تشکیل می‌شود، میزان تعامد زیرفضای علامت و نوفه در زاویه‌های مختلف را می‌توان با استفاده از رابطه زیر محاسبه نمود:

$$f(\theta, \phi) = \frac{1}{\|U_n^H a(\theta, \phi)\|^2} \quad (6)$$

$f(\theta, \phi)$ تابع چگالی طیف فضایی الگوریتم موزیک نام دارد. در الگوریتم موزیک، زاویه‌های ورود با استفاده از جستجو روی زاویه‌های مختلف و یافتن قله‌های محلی تابع $f(\theta, \phi)$ تخمین زده می‌شوند.

۲-۲-۲. جهت‌یابی با استفاده از الگوریتم‌های مبتنی بر شکل‌دهی پرتو

دسته دیگر از الگوریتم‌های جهت‌یابی که در کاربردهای عملی مورد توجه هستند، روش‌های مبتنی بر شکل‌دهی پرتو نام دارند. در این الگوریتم‌ها با استفاده از ضرب یک بردار وزن در علامت دریافتی زاویه‌های ورود تخمین زده می‌شود. در شکل ۲ بلوک دیاگرام جهت‌یابی با استفاده از الگوریتم‌های مبتنی بر شکل‌دهی پرتو نشان داده شده است. خروجی شکل‌دهنده پرتو در حوزه بسامد که با

در رابطه (۲)، c ، سرعت انتشار صوتی، θ_k زاویه افقی علامت k -ام و ϕ_k زاویه عمودی این علامت می‌باشند. ماتریس کواریانس علامت دریافت‌شده در رابطه (۱) به صورت زیر محاسبه می‌شود:

$$R_x = E\{x(t)x^H(t)\} \quad (3)$$

$$= A(\theta, \phi)R_s A^H(\theta, \phi) + \sigma^2 I_M$$

در این رابطه، $R_s = E\{s(t)s^H(t)\}$ ماتریس کواریانس منابع را نشان می‌دهد. در عمل امکان محاسبه دقیق ماتریس کواریانس علامت دریافتی وجود ندارد. بنابراین، ماتریس کواریانس به کمک تعداد مشاهده دریافتی محدود، تخمین زده می‌شود. با فرض در اختیار بودن N مشاهده از علامت دریافتی، ماتریس کواریانس از رابطه زیر تخمین زده می‌شود.

$$\hat{R}_x = \frac{1}{N} \sum_{t=1}^N x(t)x^H(t) \quad (4)$$

۲-۲. الگوریتم‌های جهت‌یابی

برای جهت‌یابی علامت، راهکارهای مختلفی پیشنهاد شده است که از مهم‌ترین آن‌ها می‌توان به روش بیش‌ترین شباهت، روش‌های مبتنی بر زیر فضا و روش‌های مبتنی بر شکل‌دهی پرتو اشاره کرد.

روش بیش‌ترین شباهت، یک روش بهینه بوده و در شرایط مختلف اعم از تعداد مشاهده کم، نسبت توان علامت به توان نوفه پایین و همبسته بودن منابع، کارایی مناسبی دارد. به دلیل پیچیدگی بسیار بالا، در عمل الگوریتم بیش‌ترین شباهت مورد استفاده قرار نمی‌گیرد. روش‌های مبتنی بر زیرفضا و شکل‌دهی پرتو، روش‌های زیربهینه‌ای هستند که دارای عملکرد مناسبی می‌باشند. از مهم‌ترین الگوریتم‌های زیرفضا می‌توان به الگوریتم موزیک اشاره کرد. این الگوریتم قابلیت اعمال به علامت دریافتی هرگونه آرایه‌ای را دارد. الگوریتم‌های دی‌ای‌اس و ام‌وی‌دی‌آر مهم‌ترین و شناخته‌شده‌ترین دسته از روش‌های مبتنی بر شکل‌دهی پرتو هستند. این الگوریتم‌ها نیز کارایی نسبتاً خوبی دارند. در ادامه این سه روش جهت‌یابی مورد بررسی قرار می‌گیرند.

۲-۲-۱. جهت‌یابی با استفاده از الگوریتم موزیک

$$w_{\text{MVDR}}(\theta, \phi) = \frac{R^{-1}(\omega) a(\theta, \phi)}{a^H(\theta, \phi) R^{-1}(\omega) a(\theta, \phi)} \quad (11)$$

در عمل، تعداد مشاهده‌های کمی از علامت در اختیار بوده که منجر به تکین شدن ماتریس همبستگی می‌شود. برای رفع این مشکل، ماتریس کواریانس و هم‌چنین کاهش حساسیت الگوریتم ام‌وی‌دی‌آر به خطای بردار جهت‌دهی، از بارگذاری روی درایه‌های قطری به صورت زیر استفاده می‌شود [۱۰]:

$$\hat{R}(\omega) = R(\omega) + I_M \cdot \varepsilon \cdot \text{trace}\{R(\omega)\} \quad (12)$$

در این رابطه، ε یک ضریب از پیش تعیین شده می‌باشد. توان خروجی شکل‌دهنده پرتو در الگوریتم ام‌وی‌دی‌آر به صورت رابطه زیر ساده می‌شود:

$$P_{\text{MVDR}}(\omega, \theta, \phi) = \frac{1}{a^H(\theta, \phi) \hat{R}^{-1}(\omega) a(\theta, \phi)} \quad (13)$$

رابطه فوق طیف توان بسامدی زاویه‌ای الگوریتم ام‌وی‌دی‌آر را نشان می‌دهد. زاویه‌های ورود منابع با استفاده از مکان قله‌های این تابع تخمین زده می‌شوند.

۲-۳. واحد پردازنده‌ی گرافیکی

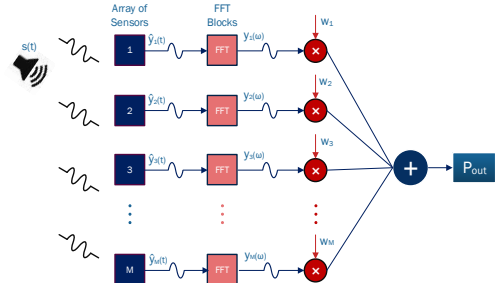
ویژگی اصلی جی‌پی‌یوها تعداد هسته‌های پردازشی آن‌ها است. جی‌پی‌یو از هسته‌های پردازشی بسیار بیش‌تری نسبت به یک جی‌پی‌یو تشکیل شده است که این ویژگی، قابلیت پردازش موازی را به جی‌پی‌یوها می‌دهد. اگرچه سی‌پی‌یوها از هسته‌های پردازشی قوی‌تری بهره‌مند هستند، در شرایطی که هدف انجام عملیاتی با محاسبات کم به صورت مکرر باشد، استفاده از جی‌پی‌یو و پردازش موازی از نظر زمان و سرعت انجام عملیات بسیار به صرفه‌تر است.

۲-۳-۱. الگوی برنامه‌نویسی کودا

کودا بستری برای محاسبات موازی همه‌منظوره است که رابط‌های برنامه‌نویسی را در اختیار برنامه‌نویسان قرار می‌دهد و به طراحان نرم‌افزار اجازه می‌دهد تا از توانایی‌های جی‌پی‌یوهای ساخته‌شده توسط شرکت انویدیا در جهت محاسبات همه‌منظوره روی واحد پردازش گرافیکی استفاده کنند. این بستر به گونه‌ای طراحی شده است که با زبان‌های برنامه‌نویسی C، C++، فورتن و متلب کار می‌کند [۱۱]. شیوه برنامه‌نویسی کودا به گونه‌ای است که امکان برنامه‌نویسی روی هر دو پردازنده جی‌پی‌یو و

$B(\omega, \theta, \phi)$ نشان داده می‌شود، طبق رابطه زیر به دست می‌آید:

$$B(\omega, \theta, \phi) = w^H y = \sum_{m=1}^M w_m^*(\theta, \phi) y_m(\omega) \quad (7)$$



شکل ۲ یک نمایش ساختار شکل‌دهنده پرتو در حوزه بسامد [۷].

در الگوریتم شکل‌دهی پرتوی کلاسیک یا دی‌ای‌اس، از محاسبه توان خروجی شکل‌دهنده پرتو در زاویه‌های مختلف برای تخمین راستای منبع علامت دریافتی استفاده می‌شود [۷]. توان خروجی شکل‌دهنده پرتو در حالت کلی، طبق رابطه زیر به دست می‌آید:

$$P_{\text{out}}(\theta, \phi) = E(|B(w, \theta, \phi)|^2) = w^H(\theta, \phi) R(\omega) w(\theta, \phi) \quad (8)$$

در رابطه فوق، $R(\omega)$ ماتریس کواریانس مکانی آرایه می‌باشد و به صورت رابطه زیر تعریف می‌شود:

$$R(\omega) = E(y(\omega) y^H(\omega)) \quad (9)$$

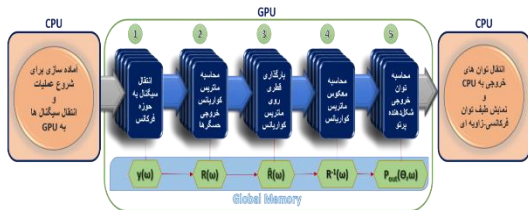
الگوریتم دی‌ای‌اس یک روش شکل‌دهی پرتو غیروفاقی می‌باشد، در این الگوریتم بردار وزن‌ها برابر بردار جهت‌دهی بهنجار شده بوده و طبق رابطه (۱۰) تعریف می‌شود:

$$w_{\text{DAS}}(\theta, \phi) = \frac{a(\theta, \phi)}{\|a(\theta, \phi)\|} \quad (10)$$

نشان داده شده است که این روش برای حالتی که تعداد منابع بیش‌تر از یک باشد، کارایی مناسبی ندارد [۸].

در روش ام‌وی‌دی‌آر، بردار وزن‌ها متناسب با علامت‌های دریافتی آرایه انتخاب می‌شوند. در این الگوریتم، وزن‌ها به گونه‌ای انتخاب می‌شوند که توان خروجی شکل‌دهنده کمینه شده، به شرطی که توان علامت اصلی در خروجی تضعیف نشود. با حل این مسئله کمینه‌سازی توسط روش ضرایب لاگرانژ، بردار وزن بهینه الگوریتم ام‌وی‌دی‌آر به صورت زیر تعیین می‌شود [۹]:

روی ممتلب نیز به‌طور کامل و یکسان پیاده‌سازی شده‌اند.



شکل ۳ نمودار بلوکی پیاده‌سازی ام‌وی‌دی‌آر روی واحد پردازنده گرافیکی.

۴. نتایج

برای شبیه‌سازی، سرعت انتشار علامت ۳۴۰ متر بر ثانیه، تعداد حسگرهای آرایه چهار، نرخ نمونه‌برداری ۹۶ کیلوهرتز، زمان هر پنجره جهت‌یابی شده ۲۰۰ میلی‌ثانیه، تعداد بخش‌ها در هر پنجره ۱۰، تعداد نمونه در هر بخش ۱۹۲۰، بازه‌ی بسامدی جستجو ۱۰۰ تا ۳۰۰۰ هرتز، تعداد مولفه‌های بسامدی برای جستجوی پنج مولفه بزرگ، بازه‌ی زاویه‌ای افقی جستجو شده صفر تا ۳۵۹ درجه، بازه‌ی زاویه‌ای عمودی جستجو شده صفر تا ۹۰ درجه و مختصات قرارگیری حسگرها در آرایه به‌صورت زیر در نظر گرفته می‌شود.

$$\begin{bmatrix} 1,006,01,37; -0,5015,0,8686,1,37; \\ -0,5015, -0,8686,1,37; 0,2,392 \end{bmatrix}$$

سه الگوریتم شناخته‌شده‌ی دی‌ای‌اس، ام‌وی‌دی‌آر و موزیک در نرم‌افزار ممتلب به‌صورت سریال و در نرم‌افزار ویژوال استودیو^۴ به کمک الگوی کودا به‌صورت موازی پیاده‌سازی شدند. این پیاده‌سازی‌ها در هر دو نرم‌افزار کاملاً مشابه و یکسان انجام شده به گونه‌ای که ماتریس توان خروجی الگوریتم در هر دو برنامه کاملاً برابر خواهند شد. ابتدا جهت مشاهده‌ی طیف توان خروجی هر سه الگوریتم، علامتی از یک منبع با زاویه‌های افقی و عمودی ۲۰۰ و ۵۰ درجه نسبت به آرایه حسگرها و بسامد ۸۰۰ هرتز به همراه نوفه سفید با SNR=۵dB شبیه‌سازی می‌شود.

نمودارهای شکل‌های ۴ تا ۶ به‌ترتیب نتایج حاصل از الگوریتم‌های دی‌ای‌اس، ام‌وی‌دی‌آر و موزیک را نمایش

می‌دهد؛ قسمت‌های سریال برنامه روی سی‌پی‌یو اجرا می‌شود، سپس برنامه به جی‌پی‌یو منتقل شده و پس از اجرای عملیات مربوط به قسمت‌های موازی، دوباره به سی‌پی‌یو باز می‌گردد. بخش‌های موازی برنامه در یک تابع کرنل^۱ نوشته می‌شوند و هر کرنل توسط تعداد زیادی ترد^۲ به‌صورت موازی اجرا می‌شود. سازمان‌دهی تردها در کودا توسط بلوک‌ها و گریدها^۳ انجام می‌شود. هر بلوک شامل یک آرایه سه-بعدی از تردها است بنابراین هر ترد در هر بلوک توسط موقعیت مکانی (x, y, z) شناخته می‌شود. در فراخوانی هر کرنل، برنامه‌نویس می‌تواند تعداد بلوک‌ها و تعداد تردها در هر بلوک را برای اجرای آن کرنل انتخاب نماید.

۲-۳-۲. پیاده‌سازی الگوریتم‌های جهت‌یابی روی جی‌پی‌یو

یک ویژگی مهم الگوریتم‌های شکل‌دهنده پرتو و موزیک، قابلیت موازی‌سازی آن‌ها است. به این ترتیب که می‌توان توان خروجی شکل‌دهنده پرتو در هر زاویه و بسامد خاص را به‌طور مستقل محاسبه کرد. در نتیجه به کمک جی‌پی‌یو و پردازش موازی، می‌توان سرعت انجام محاسبات را افزایش داد.

در این مقاله برای پردازش موازی الگوریتم‌های دی‌ای‌اس، ام‌وی‌دی‌آر و موزیک از الگو برنامه‌نویسی کودا و زبان C++ استفاده می‌شود. شکل ۳، طرح کلی ساختار طراحی شده به منظور پیاده‌سازی الگوریتم ام‌وی‌دی‌آر را نشان می‌دهد. همان‌طور که اشاره شد، در کودا ابتدا برنامه روی سی‌پی‌یو اجرا می‌شود تا دسترسی لازم به قسمت‌های موازی فراهم گردد. در این بخش، ابتدا بازه بسامدی و زاویه‌ای موردنظر برای جهت‌یابی منبع علامت تعیین می‌شود، سپس علامت‌های همه حسگرها به‌صورت یک ماتریس به حافظه گلوبال جی‌پی‌یو منتقل می‌شوند. پنج بخش اصلی که برای پیاده‌سازی ام‌وی‌دی‌آر، تقریباً با اندک تغییری در تعدادی از بخش‌ها برای الگوریتم‌های دی‌ای‌اس و موزیک نیز استفاده می‌شوند. هم‌چنین این الگوریتم‌ها علاوه بر کودا،

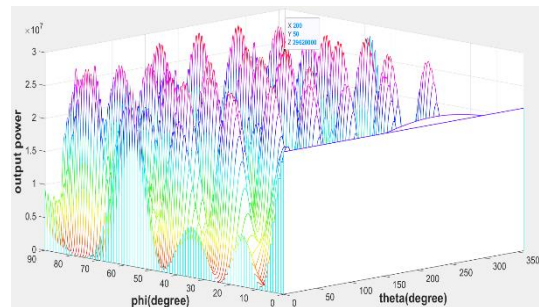
¹ Kernel
² Thread
³ Grid

⁴ Visual Studio

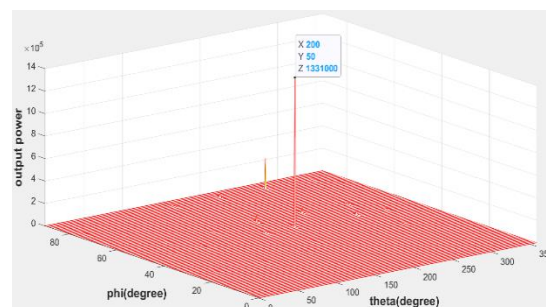
همان‌طور که نتایج نشان می‌دهند، الگوریتم‌های ام‌وی‌دی‌آر و موزیک از دقت بسیار بالاتری برخوردار هستند. با توجه به نمودارهای این دو الگوریتم، مشاهده می‌شود که الگوریتم ام‌وی‌دی‌آر دارای قله‌ی توان چندین برابر بزرگ‌تر از الگوریتم‌های دیگر می‌باشد. در این الگوریتم در سایر زاویه‌ها نیز قله‌هایی (هرچند کوچک) مشاهده می‌شود که این قله‌ها در الگوریتم موزیک به خوبی حذف شده‌اند. با بررسی دقیق‌تر توان‌ها در دو الگوریتم مشاهده می‌شود که نسبت قله به زمینه در الگوریتم موزیک تقریباً ۲٫۵ برابر این نسبت در الگوریتم ام‌وی‌دی‌آر می‌باشد که نشان از دقت چند برابری الگوریتم موزیک نسبت به الگوریتم ام‌وی‌دی‌آر دارد. اگر این خطاهای جزئی کنار گذاشته شوند، در مجموع هر دو الگوریتم توانسته‌اند زاویه منبع شبیه‌سازی شده را به درستی و با دقت بسیار بالایی تخمین بزنند. همان‌طور که گفته شد، نمودارهای خروجی حاصل از اجرای الگوریتم روی متلب و کودا کاملاً با یکدیگر برابر هستند.

حال دو منبع علامت دیگر در زاویه‌های افقی ۳۰۰ و ۱۰۰ و عمودی ۸۰ و ۲۰ درجه با بسامدهای ۲۰۰ و ۱۵۰۰ هرتز در کنار منبع قبلی شبیه‌سازی می‌شوند. الگوریتم دی‌ای‌اس به علت دقت پایین، قادر به تخمین جهت دو منبع با استفاده از تنها چهار حسگر نیست. اما دو الگوریتم دیگر تا سه منبع را می‌توانند به راحتی تخمین بزنند. نتایج حاصل از اعمال این الگوریتم‌ها روی علامت شبیه‌سازی شده به صورت نمودارهای شکل‌های ۷ و ۸ می‌باشند. با مقایسه‌ی این نمودارها این نتیجه حاصل می‌شود که الگوریتم موزیک در حذف نوفه، بسیار بهتر از ام‌وی‌دی‌آر عمل می‌کند، اما این بار نیز هر دو الگوریتم توانسته‌اند با دقت بالایی زوایای منابع شبیه‌سازی شده را به درستی تخمین بزنند. نکته‌ی جالب توجه در این الگوریتم‌ها، تاثیر نوفه سفید بر دقت جهت‌یابی آن‌ها می‌باشد تا به اینجا مقدار اس‌ان‌آر^۱ روی مقدار ۵ دسی‌بل ثابت بوده است، حال اگر اس‌ان‌آر به ۲۰ دسی‌بل افزایش یابد، نتیجه‌ی خروجی الگوریتم ام‌وی‌دی‌آر به صورت نمودار شکل ۹ خواهد شد که در مقایسه با نمودار شکل ۷،

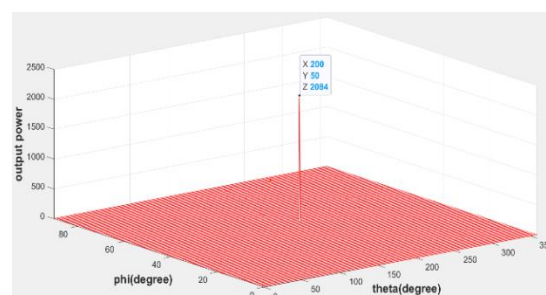
می‌دهند که در آن‌ها محور عمودی، توان خروجی شکل‌دهنده پرتو و محورهای افقی شامل زاویه افقی (θ) و زاویه عمودی (ϕ) می‌باشند. از نمودار مربوط به الگوریتم دی‌ای‌اس مشخص است که این الگوریتم نسبت به دو الگوریتم دیگر در جستجوی دو-بعدهی دقت بسیار پایین‌تری دارد. علت این مسئله، پایه‌ای بودن این الگوریتم و ساده‌تر بودن عملیات انجام گرفته در آن نسبت به دو الگوریتم دیگر می‌باشد. هم‌چنین، در بخش‌های قبل گفته شد که این الگوریتم یک شکل‌دهنده پرتو غیروفاقی است و هیچ اصلاحی بر بردار وزن‌های خود متناسب با علامت دریافتی در جهت بهبود دقت انجام نمی‌دهد. با این حال این الگوریتم توانسته زاویه‌ی شبیه‌سازی شده را به درستی پیدا کند.



شکل ۴ طیف توان خروجی الگوریتم دی‌ای‌اس برای یک منبع.

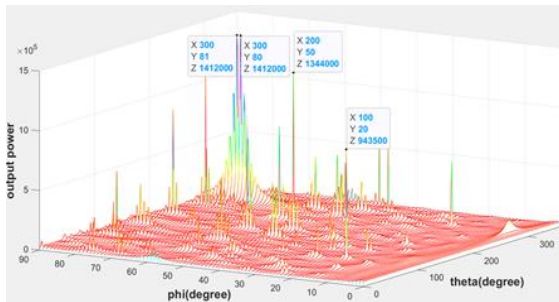


شکل ۵ طیف توان خروجی الگوریتم ام‌وی‌دی‌آر برای یک منبع.



شکل ۶ طیف توان خروجی الگوریتم موزیک برای یک منبع.

^۱ SNR



شکل ۱۰ طیف توان خروجی الگوریتم اموی دی آر برای سه منبع در SNR=-5dB

یکی از اهداف مهم این مقاله، کاهش زمان اجرای جهت یابی و رسیدن به حالت زمان حقیقی بوده است. زمان لازم برای جهت یابی علامت روی هر پنجره داده، به عواملی از قبیل سخت افزار مورد استفاده، تعداد حسگرها و پهنای باند بسامدی (متناسب با آن تعداد مولفه های بسامدی جستجو شده) وابسته است. این الگوریتم روی سخت افزاری با مشخصات زیر پایاده سازی شده است:

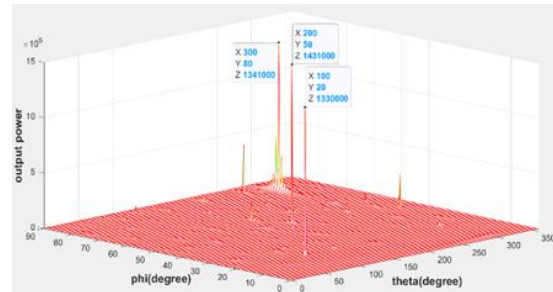
سی پی یو: اینتل کور آی ۷-۱۷۷۰۰^۱ کیو

جی پی یو: نویدیا جفورس جی تی ایکس ۱۰۶۰^۲

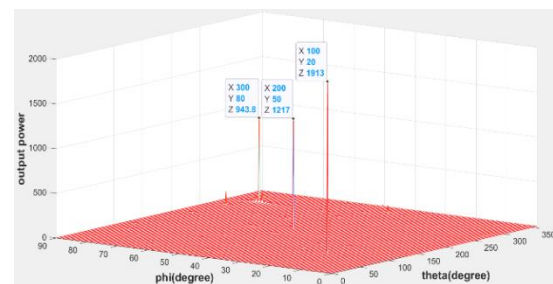
این الگوی جی پی یو طبق اطلاعات شرکت نویدیا^۳ دارای ۱۲۸۰ هسته کودا بوده و قدرت بالایی در پردازش موازی دارد. سی پی یو مورد استفاده نیز دارای چهار هسته و هشت ترد می باشد. تعداد حسگرها در این مقاله مطابق با تجهیزات موجود، برابر چهار حسگر در نظر گرفته شده اند. همان طور که گفته شد این تعداد حسگر تنها قادر به تشخیص بیشینه سه منبع می باشد. اما از طرف دیگر، تعداد کم حسگرها منجر به افزایش سرعت جهت یابی خواهد شد.

در این بخش از شبیه سازی، تاثیر این متغیر بر سرعت جهت یابی بررسی خواهد شد. متغیر دیگر تاثیرگذار بر روی این مسئله، تعداد مولفه های بسامدی جستجو شده می باشد. به این ترتیب که هرچه تعداد مولفه های جستجو شده بیش تر شود، زمان اجرای الگوریتم ها نیز بیش تر خواهد شد. در این مقاله به منظور کاهش زمان اجرا

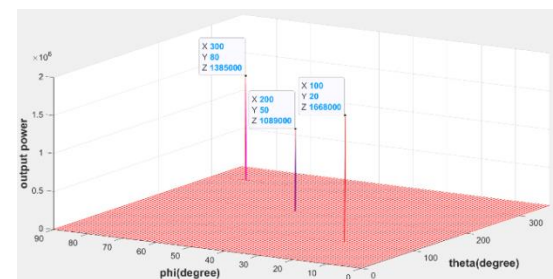
افزایش دقت محسوسی در اثر افزایش مقدار اسان آر مشاهده می شود. این تاثیر در الگوریتم موزیک نیز به همین صورت می باشد.



شکل ۷ طیف توان خروجی الگوریتم اموی دی آر برای سه منبع در SNR=5dB



شکل ۸ طیف توان خروجی الگوریتم موزیک برای سه منبع در SNR=5dB



شکل ۹ طیف توان خروجی الگوریتم اموی دی آر برای سه منبع در SNR=20dB

در شکل ۱۰، نتیجه خروجی اموی دی آر در اسان آر برابر با ۵- دسی بل رسم شده است. نتیجه به دست آمده نشان می دهد که کاهش اسان آر، منجر به قله های متعددی در شکل شده است.

¹ CPU: Intel Core i7-7700HQ

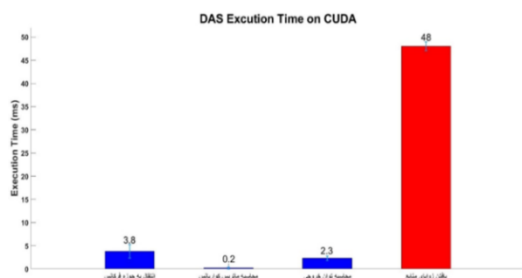
² GPU: NVIDIA GeForce GTX 1060

³ Nvidia

جهت‌یابی تفاوت‌چندانی از نظر زمان اجرا وجود ندارد (به خصوص در کودا). در کودا این الگوریتم‌ها با زمان خیلی کم‌تر (در حدود ۶ تا ۷ میلی‌ثانیه) به کار خود پایان می‌دهند. به همین علت، تفاوت زمان اجرا در آن‌ها در حدود ۱ تا ۳ میلی‌ثانیه است که این اختلاف قابل چشم‌پوشی نیست. اما باقی این زمان (حدود ۴۷ تا ۴۹ میلی‌ثانیه) برای یافتن زاویه‌ها با استفاده از بیشینه‌توان خروجی الگوریتم‌ها صرف می‌شود که به دلیل بزرگ‌بودن ماتریس توان خروجی و قابلیت موازی‌سازی بسیار پایین این روند، در کودا شدیداً دچار افت سرعت می‌گردد. به حدی که زمان صرف‌شده برای این یک مرحله، حدود هفت برابر زمان صرف‌شده برای اجرای الگوریتم‌ها از ابتدا تا انتها می‌باشد. نکته‌ی جالب توجه این است که این مرحله در متلب به علت پردازش ماتریسی این نرم‌افزار، تنها ۱ تا ۲ میلی‌ثانیه به طول می‌انجامد و باقی زمان اجرا صرفاً مربوط به عملیات‌های پردازشی خود الگوریتم‌ها است.

به‌منظور مشخص‌تر شدن این مسئله، زمان اجرای تک‌تک مراحل پردازشی هر سه الگوریتم در کودا به‌طور مجزا در نمودارهای شکل ۱۲ تا شکل ۱۴ رسم شده‌اند. در این نمودارها، تفاوت زمان اجرای مرحله‌ی یافتن زوایای منابع، با زمان اجرای سایر بخش‌های الگوریتم‌ها که در واقع بخش‌های اصلی جهت‌یابی نیز می‌باشند، به‌وضوح مشخص است.

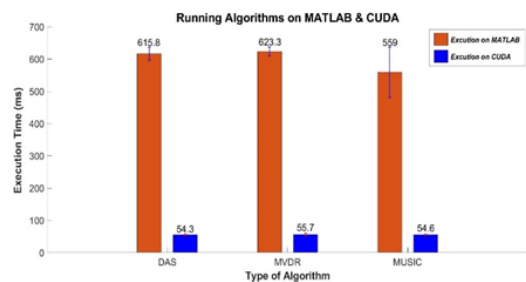
شکل ۱۲ زمان اجرای مراحل مختلف الگوریتم دی‌ای‌اس را نشان می‌دهد.



شکل ۱۲ زمان اجرای بخش‌های مختلف الگوریتم دی‌ای‌اس در کودا.

این الگوریتم پایه‌ی دو الگوریتم دیگر می‌باشد و به این علت تعداد مراحل کم‌تری برای پردازش دارد. حال با وجود این نمودار و با حذف زمان مربوط به مرحله‌ی یافتن

و حذف عملیات‌های اضافی، مولفه‌های بسامدی در بازه‌ی ۱۰۰ تا ۳۰۰۰ هرتز به‌صورت صعودی مرتب شده، سپس بزرگ‌ترین مولفه‌ها انتخاب و الگوریتم جهت‌یابی روی آن‌ها پیاده شده است. به‌صورت پیش فرض تعداد این مولفه‌ها پنج در نظر گرفته شدند. الگوریتم‌ها طبق پیش‌فرض‌هایی که در بخش‌های قبل برای متغیرها بیان شد روی متلب و کودا اجرا گردیدند. زمان هر پنجره (که به ۱۰ بخش تقسیم شده است) برابر ۲۰۰ میلی‌ثانیه و طبق بسامد نمونه‌برداری ۹۶ کیلوهرتز، شامل ۱۹۲۰۰ نمونه از علامت می‌باشد. بنابراین، به‌منظور رسیدن به حالت زمان حقیقی، زمان اجرای الگوریتم‌ها می‌بایست کم‌تر از ۲۰۰ میلی‌ثانیه باشد. نتایج زمان اجرای الگوریتم‌ها روی دو نرم‌افزار در نمودار شکل ۱۱ آمده است.

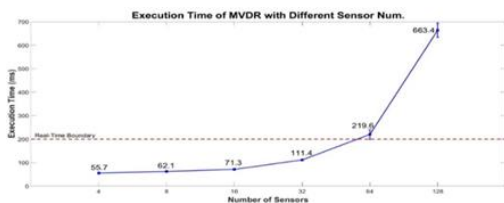


شکل ۱۱ زمان اجرای الگوریتم‌ها روی متلب و کودا بر حسب میلی‌ثانیه.

نمودار شکل ۱۱ به وضوح قدرت پردازش موازی را در برابر پردازش سریال نشان می‌دهد. سرعت اجرا روی کودا در تمامی الگوریتم‌ها تا بیش از ۱۰ برابر نسبت به متلب افزایش یافته است. اجرای این الگوریتم‌ها به‌صورت سریال، به‌هیچ‌وجه زمان حقیقی نبوده اما به کمک پردازش موازی زمان اجرای این الگوریتم‌ها تا یک چهارم بیشینه زمان اجرا برای ماندن در حالت زمان حقیقی کاهش داده شد. این مسئله علاوه بر رساندن روند جهت‌یابی علامت به‌حالت زمان حقیقی، فرصت خوبی را برای افزایش دقت جهت‌یابی به کمک افزایش حسگرها، یا جستجوی منابع بیش‌تر در محیط به کمک افزایش مولفه‌های بسامدی و یا استفاده از سخت‌افزار ضعیف‌تر با هدف کاهش قیمت سامانه جهت‌یابی، بدون نگرانی خارج‌شدن از حالت زمان حقیقی ایجاد می‌کند. در نمودار شکل ۱۱ مشاهده می‌شود که بین سه الگوریتم

موازی‌سازی بالای این بخش، به خوبی توانسته است روی کودا بهینه شده و زمان اجرا را شدیداً کاهش دهد. اما دلیل اینکه مرحله اول از مرحله محاسبه‌ی توان نیز بیش‌تر زمان صرف کرده است مربوط به عملیات مرتب‌کردن در این بخش می‌باشد (به‌منظور یافتن بزرگ‌ترین مولفه‌های بسامدی) که این عملیات (همانند مرحله یافتن زاویه‌ها) قابلیت موازی‌سازی بسیار کمی دارد و زمان زیادی را برای اجرا روی کودا صرف می‌کند.

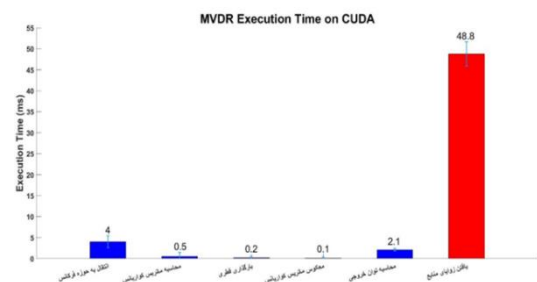
در آخر، به‌منظور نمایش میزان قابلیت افزایش دقت الگوریتم با این سخت‌افزار و بدون خارج‌شدن از حالت زمان حقیقی، با ثابت در نظر گرفتن تعداد مولفه‌های بسامدی جستجو شده روی عدد پنج (همان‌طور که در اجراهای قبلی نیز چنین بود)، تعداد حسگرها از چهار تا ۱۲۸ عدد تغییر یافتند. این عمل روی الگوریتم ام‌وی‌دی‌آر که نسبت به سایر الگوریتم‌ها زمان اجرای بیش‌تری را صرف می‌کرد، پیاده می‌شود. برای سادگی آرایه‌ی حسگرها نیز به‌صورت یک آرایه‌ی خطی در نظر گرفته می‌شوند. نتایج حاصل از این عملیات در شکل ۱۵ نمایش داده شده است. نتایج شکل ۱۵، به وضوح نشان می‌دهند که الگوریتم ام‌وی‌دی‌آر روی این سخت‌افزار، تا تعداد ۳۲ حسگر برای آرایه را می‌تواند به خوبی به‌صورت زمان-حقیقی پردازش کند. خط‌چین قرمز نشان‌دهنده‌ی مرز حالت زمان حقیقی می‌باشد. همان‌طور که در ابتدای این بخش گفته شد، زمان اجرا علاوه بر تعداد حسگر، به قدرت سخت‌افزار مورد استفاده و تعداد مولفه‌های بسامدی جستجو شده نیز بستگی دارد. در نتیجه در صورت استفاده از سخت‌افزار ضعیف‌تر یا نیازمندی به پردازش تعداد مولفه‌ی بسامدی بیش‌تر از پنج عدد، بیشینه تعداد حسگرهای قابل استفاده برای ماندن در حالت زمان حقیقی نیز کاهش می‌یابد.



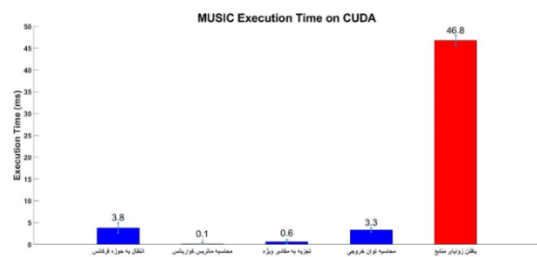
شکل ۱۵ زمان اجرای الگوریتم ام‌وی‌دی‌آر با تعداد حسگرهای مختلف بر حسب میلی‌ثانیه.

زوایای منابع، به‌راحتی می‌توان تفاوت زمان اجرای دی‌ای‌اس با دو الگوریتم دیگر را مشاهده کرد. اما این اندک اختلاف در زمان اجرا، در ازای دقت بسیار پایین‌تر این الگوریتم به‌دست آمده است که به هیچ‌وجه به صرفه نیست.

مراحل اضافی که در الگوریتم ام‌وی‌دی‌آر نسبت به دی‌ای‌اس اجرا می‌شوند، مرحله‌ی بارگذاری قطری روی ماتریس کواریانس (به‌منظور رفع مشکل تکینگی) و محاسبه‌ی معکوس ماتریس کواریانس می‌باشند. در حالی که این مراحل اضافی در الگوریتم موزیک تنها مربوط به مرحله‌ی تجزیه‌ی ماتریس کواریانس به مقادیر ویژه می‌باشند. با مشاهده‌ی نمودار این دو الگوریتم، این نتیجه حاصل می‌شود که این دو از نظر زمان اجرا تفاوت خاصی با یکدیگر ندارند.



شکل ۱۳ زمان اجرای بخش‌های مختلف الگوریتم ام‌وی‌دی‌آر در کودا.



شکل ۱۴ زمان اجرای بخش‌های مختلف الگوریتم موزیک در کودا.

مقایسه‌ی نمودارها نشان می‌دهد که بیش‌ترین هزینه‌ی زمانی در این الگوریتم‌ها (با در نظر نگرفتن زمان مرحله‌ی آخر) مربوط به مراحل انتقال به حوزه بسامد و محاسبه‌ی توان خروجی می‌باشند. مهم‌ترین و سنگین‌ترین عملیات‌های پردازشی در این الگوریتم‌ها در مرحله‌ی محاسبه‌ی توان انجام می‌شوند که در متلب نیز بیش‌ترین زمان اجرا در این بخش صرف می‌شد. اما به دلیل قابلیت

۵. بحث و نتیجه‌گیری

پیچیدگی محاسباتی زیاد الگوریتم‌های جهت‌یابی مانع از اجرای زمان‌حقیقی این عملیات در زبان‌های برنامه‌نویسی سریال می‌شود. این در حالی است که یکی از شبه‌سنج‌های مهم هر سامانه جهت‌یابی قابلیت جهت‌یابی زمان‌حقیقی آن می‌باشد. برای افزایش سرعت اجرای این الگوریتم‌ها می‌توان از پیاده‌سازی سخت‌افزاری (استفاده از اف‌پی‌جی‌ای^۱ و یا دی‌اس‌پی^۲) و یا برنامه‌نویسی موازی به کمک قدرت جی‌پی‌یو استفاده نمود. اگرچه پیاده‌سازی سخت‌افزاری سرعت بالاتری نسبت به برنامه‌نویسی موازی دارد اما منجر به کاهش انعطاف‌پذیری برنامه خواهد شد؛ بنابراین در این مقاله از رویکرد دوم (برنامه‌نویسی موازی) به منظور پیاده‌سازی الگوریتم‌های دی‌اس‌پی، ام‌وی‌دی‌آر و موزیک استفاده شده است. زمان‌برترین عملیات پردازشی موجود در این الگوریتم‌ها در حالت سریال، محاسبه توان خروجی شکل‌دهنده پرتو می‌باشد؛ زیرا این عملیات به ازای هر زاویه‌ی افقی و عمودی و هر مولفه‌ی بسامدی یک‌بار تکرار می‌گردد. اما به علت قابلیت موازی‌سازی بالای این عملیات، زمان اجرای این بخش به وسیله‌ی الگوی برنامه‌نویسی کودا به شدت کاهش یافت. نتایج ارائه‌شده در این مقاله نشان می‌دهند که با استفاده از کارت گرافیکی موجود و الگوی برنامه‌نویسی کودا می‌توان سرعت اجرای الگوریتم را نسبت به برنامه‌نویسی سریال تا بیش از ۱۰ برابر افزایش داد. با بررسی زمان اجرای بخش به بخش این الگوریتم‌ها مشخص شد، زمانی که به منظور یافتن زوایای منابع از ماتریس توان خروجی در کودا صرف می‌شود، تقریباً هفت برابر زمانی است که برای اجرای بخش‌های اصلی الگوریتم جهت‌یابی صرف می‌شود؛ که علت آن قابلیت پایین موازی‌سازی در این عملیات و عدم وجود قدرت پردازش ماتریسی در کودا می‌باشد. همچنین تأثیر عامل تعداد حسگرها در آرایه نیز روی سرعت اجرای برنامه بررسی گردید و مشاهده شد که این الگوریتم‌ها روی سخت‌افزار مورد استفاده در این مقاله، قابلیت اجرای زمان‌حقیقی عملیات جهت‌یابی با بیشینه ۳۲ حسگر را دارا می‌باشند.

۷. تقدیر و تشکر

نویسندگان از حمایت انجمن علوم صوتی ایران و پژوهشکده صوتیات تشکر می‌کنند.

۸. تضاد منافع

این مطالعه تضاد منافع ندارد.

فهرست منابع

- [1] J. Benesty, J. Chen, Y. Huang, "Conventional beamforming techniques," in *Microphone Array Signal Processing*, vol. 1, 2008.
- [2] F. Vignon, M. R. Burcher, "Capon beamforming in medical ultrasound imaging with focused beams," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 55, no. 3, pp. 619-628, 2008.
- [3] F. Yan, M. Jin, X. Qiao, "Low-complexity DOA estimation based on compressed MUSIC and its performance analysis," *IEEE Transactions on Signal Processing*, vol. 61, no. 8, pp. 1915-1930, 2013.
- [4] X. Wu, W. Zhu, J. Yan, "Direction of Arrival Estimation for Off-Grid Signals Based on Sparse Bayesian Learning," *IEEE Sensors of Journal*, vol. 16, no. 7, 2016.
- [5] J. Dai, H. Cheung, "Sparse bayesian learning approach for outlier-resistant direction-of-arrival estimation," *IEEE Transactions on Signal Processing*, vol. 66, no. 3, pp. 744-756, 2018.
- [6] Z. Yang, J. Li, P. Stoica, L. Xie, "Sparse methods for direction-of-arrival estimation," *Academic Press Library in Signal Processing*, vol. 7, pp. 509-581, 2018.
- [7] L. Liang, Y. Shi, Y. Shi, Z. Bai, W. He, "Two-dimensional DOA estimation method of acoustic vector sensor array based on sparse recovery," *Digital Signal Processing*, vol. 120, pp. 103294, 2022.
- [8] J. Grythe, A.S. Norsonic, "Beamforming algorithms-beamformers," *Technical Note, Norsonic AS*, 2015.
- [9] S.-G. Shi, Y. Li, Z.-R. Zhu, J. Shi, "Real-valued robust DOA estimation method for uniform circular acoustic vector sensor arrays based on worst-case performance optimization," *Applied Acoustics*, vol. 148, pp. 495-502, 2019.
- [10] A. Hakam, R.M. Shubair, E. Salahat, "Enhanced DOA Estimation Algorithms Using MVDR and MUSIC," in *2013 International Conference on Current Trends in Information Technology (CTIT)*, 2013.

¹ FPGA

² DSP

- [11] Y. Fathi, A. Mahloojifar, B. Mohammadzadeh Asl, "Real-time implementation of adaptive beam-forming in medical ultrasound imaging by means of GPU-parallel processing," *Journal of Acoustical Society of Iran*, vol. 1, no. 1, pp. 51-60, 2014.