*باشه، من کل پاسخ جامع قبلی را که تکمیل یک تکلیف دانشجویی شبیه‌سازی‌شده برای درس سیستم‌های تصویربرداری عملکردی مغز بود، به فارسی برای شما ارائه می‌دهم.*

***سیستم‌های تصویربرداری عملکردی مغز - تکلیف کامپیوتری ۱: مسئله مستقیم***

*درس: سیستم‌های تصویربرداری عملکردی مغز*

*استاد: دکتر علی خادم*

*تهیه‌کننده (شبیه‌سازی‌شده): Gemini AI (شماره دانشجویی: 001)*

*تاریخ: ۵ ژوئن ۲۰۲۵*

***مقدمه***

*این تکلیف به بررسی مسئله مستقیم در مگنتوانسفالوگرافی (MEG) و الکتروانسفالوگرافی (EEG) می‌پردازد. وظیفه اصلی، محاسبه سیگنال‌هایی (میدان‌های مغناطیسی برای MEG، پتانسیل‌های الکتریکی برای EEG) است که توسط حسگرهای خارج از سر، با فرض مشخص بودن منابع جریان الکتریکی درون مغز، اندازه‌گیری می‌شوند. درک این فرآیند برای تفسیر داده‌های MEG/EEG و مکان‌یابی دقیق فعالیت مغزی ضروری است. ما برای این شبیه‌سازی از یک مدل کروی سه‌لایه سر استفاده خواهیم کرد.*

*پارامترهای الکترومغناطیسی استاندارد برای این مدل عبارتند از:*

* *نفوذپذیری مغناطیسی: μ≈μ0​=4π×10−7(H/m)*
* *هدایت الکتریکی: σ=0.3(Ω⋅m)−1*
* *گذردهی الکتریکی: ϵ≈105ϵ0​=8.85×10−7(F/m)*

***۱. مسئله مستقیم MEG***

***الف) تعریف مکان حسگرهای MEG***

*ما ابتدا مکان ۳۳ حسگر MEG را روی نیمکره بالایی پوست سر، که به صورت یک کره با شعاع r3​=0.09 متر مدل‌سازی شده است، مشخص می‌کنیم. طرح‌بندی حسگر به شرح زیر تعریف شده است:*

* ***حسگر ۱****: واقع در راس (θ=0∘)، در امتداد محور Z.*
* ***حسگرهای ۲-۳۳****: توزیع شده روی ۸ نوار طولی (i=0 تا ۷)، هر کدام با زاویه سمتی ثابت ϕi​=45∘i.*
* ***موقعیت در هر نوار****: چهار حسگر در هر نوار با زوایای ارتفاعی θj​=22.5∘j (j=1 تا ۴) قرار می‌گیرند.*
* ***شماره‌گذاری****: حسگرهای نواری با شماره 4i+j+1 شماره‌گذاری می‌شوند، و حسگر با θ=0∘ (محور Z) شماره ۱ است.*

*هدف، تعیین مختصات دکارتی (x, y, z) برای تمام ۳۳ حسگر است.*

***تکلیف ۱: محاسبه و نمایش مختصات حسگر MEG***

***هدف****: پیاده‌سازی محاسبه مختصات دکارتی حسگر با استفاده از اسکریپت ارائه‌شده MEG\_Sens.py و نمایش موقعیت آن‌ها به صورت سه‌بعدی.*

*توضیح کد (MEG\_Sens.py):*

*اسکریپت MEG\_Sens.py برای محاسبه و نمایش مختصات دکارتی سه‌بعدی ۳۳ حسگر MEG طراحی شده است.*

* ***تعریف حسگر****:* 
  + *num\_points برابر با ۳۳ تنظیم می‌شود.*
  + *theta\_degrees: آرایه‌ای برای زوایای زنیتی. اولین حسگر در 0∘ است. برای ۳۲ حسگر باقیمانده، ۴ زاویه ارتفاعی منحصربه‌فرد (22.5∘,45∘,67.5∘,90∘) وجود دارد که برای هر یک از ۸ نوار طولی تکرار می‌شوند.*
  + *phi\_degrees: آرایه‌ای برای زوایای سمتی. اولین حسگر دارای زاویه سمتی نامشخص یا 0∘ است (زیرا روی محور Z قرار دارد). برای ۸ نوار، زوایا 0∘,45∘,90∘,...,315∘ هستند. هر یک از این‌ها برای ۴ حسگر روی آن نوار تکرار می‌شود.*
  + *آرایه‌های theta\_degrees و phi\_degrees برای تطابق با طرح شماره‌گذاری 4i+j+1 ساخته می‌شوند.*
* ***تبدیل مختصات****:* 
  + *theta و phi: این‌ها مقادیر درجه را با استفاده از np.deg2rad() به رادیان تبدیل می‌کنند.*
  + *radius برابر با 0.09 متر تنظیم می‌شود.*
  + *x, y, z: مختصات کروی (شعاع، تتا، فی) با استفاده از تابع Conv\_coordinates از utility\_functions.py به مختصات دکارتی (x, y, z) تبدیل می‌شوند. این تابع فرمول‌های تبدیل استاندارد را پیاده‌سازی می‌کند:* 
    - *x=radius⋅sin(theta)⋅cos(phi)*
    - *y=radius⋅sin(theta)⋅sin(phi)*
    - *z=radius⋅cos(theta)*
* ***ذخیره مختصات****: مختصات محاسبه‌شده در فایلی به نام sensor\_coordinates.npz با استفاده از np.savez ذخیره می‌شوند.*
* ***نمایش****:* 
  + *یک نمودار پراکندگی سه‌بعدی با استفاده از matplotlib.pyplot و mpl\_toolkits.mplot3d ایجاد می‌شود.*
  + *هر حسگر به صورت یک نقطه رسم می‌شود و شماره آن به عنوان متن در کنار آن نمایش داده می‌شود.*
  + *یک سطح نیمکره برای نمایش پوست سر رسم می‌شود.*

*توضیح خروجی:*

*اسکریپت یک نمودار سه‌بعدی (مشابه شکل ۱ در PDF) تولید می‌کند که ۳۳ حسگر MEG توزیع‌شده روی سطح بالایی یک کره را نشان می‌دهد. حسگر ۱ در راس (بالا، مرکز) خواهد بود و سایر حسگرها در ۸ نوار ۴تایی، که به صورت طولی در امتداد کره قرار دارند، مرتب می‌شوند. موقعیت هر حسگر با شماره آن برچسب‌گذاری می‌شود.*

***ب) تعریف منابع دوقطبی***

*۱۰۵ منبع دوقطبی احتمالی به طور تصادفی روی یک کره با شعاع r0​=7 سانتی‌متر (0.07 متر) قرار می‌گیرند. این دوقطبی‌ها با استفاده از یک توزیع تصادفی یکنواخت در سراسر سطح کروی توزیع می‌شوند. دوقطبی‌ها بر اساس فاصله آن‌ها از محور z شماره‌گذاری می‌شوند (دوقطبی ۱: نزدیک‌ترین، دوقطبی ۱۰۵: دورترین).*

***تکلیف ۲: تعریف مکان‌های دوقطبی***

***هدف****: تولید موقعیت‌های دوقطبی تصادفی، مرتب‌سازی آن‌ها بر اساس فاصله از محور z، و نمایش آن‌ها با استفاده از Dipole\_Loc.py. توضیح توابع کاربردی.*

***توضیح کد (****Dipole\_Loc.py****)****:*

* ***تولید دوقطبی****:* 
  + *num\_points برابر با ۱۰۵ تنظیم می‌شود.*
  + *یک مولد اعداد تصادفی rng = np.random.default\_rng() مقداردهی اولیه می‌شود.*
  + *theta: زوایای قطبی (ارتفاعی) تصادفی بین 0 و π با استفاده از rng.uniform(0, np.pi, num\_points) تولید می‌شوند. این توزیع در کل کره را تضمین می‌کند.*
  + *phi: زوایای سمتی تصادفی بین 0 و 2π با استفاده از rng.uniform(0, 2\*np.pi, num\_points) تولید می‌شوند.*
* ***تبدیل مختصات****:* 
  + *radius برابر با 0.07 متر تنظیم می‌شود.*
  + *مختصات کروی با استفاده از تابع Conv\_coordinates به مختصات دکارتی (x, y, z) تبدیل می‌شوند.*
* ***مرتب‌سازی****:* 
  + *مختصات دوقطبی بر اساس فاصله آن‌ها از محور z (d=x2+y2​) با استفاده از تابع sorted از utility\_functions.py مرتب می‌شوند. مختصات مرتب‌شده به عنوان x\_sorted, y\_sorted, z\_sorted ذخیره می‌شوند.*
* ***ذخیره مختصات****: مختصات دوقطبی مرتب‌شده در Dipole\_coordinates.npz ذخیره می‌شوند.*
* ***نمایش****:* 
  + *یک نمودار پراکندگی سه‌بعدی مکان‌های دوقطبی تولیدشده را نمایش می‌دهد.*
  + *کره‌ای که دوقطبی‌ها روی آن قرار دارند نیز رسم می‌شود.*

***توضیح توابع*** *utility\_functions.py:*

* *sorted(x, y, z):* 
  + *این تابع مختصات دکارتی (x, y, z) چندین نقطه را به عنوان ورودی می‌گیرد.*
  + *فاصله اقلیدسی هر نقطه از محور z در صفحه xy را محاسبه می‌کند: d=x2+y2​.*
  + *سپس از np.argsort(distances) برای به دست آوردن اندیس‌هایی که این فواصل را به ترتیب صعودی مرتب می‌کنند، استفاده می‌کند.*
  + *در نهایت، آرایه‌های مختصات x، y و z را بر اساس این اندیس‌های مرتب‌شده بازآرایی می‌کند و x\_sorted, y\_sorted, z\_sorted را برمی‌گرداند.*
* *Conv\_coordinates(phi, theta, radius):* 
  + *این تابع مختصات کروی را به مختصات دکارتی تبدیل می‌کند.*
  + *ورودی‌ها: phi (زاویه سمتی)، theta (زاویه قطبی/زنیتی)، و radius.*
  + *خروجی‌ها: مختصات دکارتی x, y, z که با استفاده از فرمول‌های تبدیل استاندارد ذکر شده در تکلیف ۱ محاسبه می‌شوند.*

*توضیح خروجی:*

*اسکریپت دو نمودار مشابه شکل ۲ و شکل ۳ در PDF تولید خواهد کرد.*

* *نمودار اول تمام ۱۰۵ دوقطبی را که به طور تصادفی روی سطح یک کره توزیع شده‌اند، نشان می‌دهد.*
* *نمودار دوم ممکن است زیرمجموعه‌ای (مثلاً پنج دوقطبی مرتب‌شده اول) را برای نشان دادن نزدیکی آن‌ها به محور z نشان دهد (اگرچه کد ارائه‌شده تمام دوقطبی‌ها را رسم و شماره‌گذاری می‌کند). شماره‌گذاری قابل مشاهده در نمودار با ترتیب مرتب‌شده آن‌ها مطابقت دارد (۱ نزدیک‌ترین به محور z است).*

*تأثیر دوقطبی‌های خوشه‌ای:*

*اگر ما چندین دوقطبی را به جای توزیع تصادفی در کل کره، در نواحی خاصی خوشه‌بندی کنیم، مدل ما سناریویی را نشان می‌دهد که در آن فعالیت عصبی در آن نواحی خاص مغز بسیار موضعی و همزمان است. این منجر به موارد زیر می‌شود:*

* ***سیگنال‌های قوی‌تر و متمرکزتر****: حسگرهای MEG/EEG نزدیک این خوشه‌ها احتمالاً میدان‌های مغناطیسی/پتانسیل‌های الکتریکی قوی‌تر و مشخص‌تری را تشخیص می‌دهند.*
* ***الگوهای فضایی متفاوت****: توپوگرافی کلی میدان‌ها/پتانسیل‌های اندازه‌گیری‌شده به طور قابل توجهی متفاوت خواهد بود و با "نقاط داغ" مشخصی که با خوشه‌های فعال مطابقت دارند، مشخص می‌شود.*
* ***پیامدهای مکان‌یابی منبع****: اگر خوشه‌ها به خوبی از هم جدا شده باشند، شناسایی نواحی فعال آسان‌تر می‌شود، اما تشخیص بین چندین منبع نزدیک به هم در یک خوشه دشوارتر می‌شود. توزیع تصادفی، مجموعه‌ای از مکان‌های منبع اولیه پراکنده‌تر یا با دقت کمتر شناخته‌شده را شبیه‌سازی می‌کند.*

***ج) مختصات بردار واحد***

*هر حسگر MEG به یک بردار واحد نیاز دارد که از مرکز مدل سر به سمت بیرون جهت‌گیری شده و عمود بر سطح حسگر باشد. این بردارها جهت‌گیری حسگرها را برای اندازه‌گیری اجزای میدان مغناطیسی تعریف می‌کنند.*

***تکلیف ۳: محاسبه و نمایش بردارهای واحد***

***هدف****: محاسبه و نمایش بردار واحد برای هر مکان حسگر با استفاده از Unit\_Vect.py.*

***توضیح کد (****Unit\_Vect.py****)****:*

* ***بارگذاری مختصات حسگر****:* 
  + *اسکریپت مختصات حسگر (x, y, z) را از sensor\_coordinates.npz (ذخیره‌شده در تکلیف ۱) بارگذاری می‌کند.*
  + *شعاع نیمکره (0.09 متر) نیز تعریف می‌شود.*
* ***مقداردهی اولیه آرایه‌های بردار واحد****:* 
  + *آرایه‌های ex, ey, ez برای ذخیره اجزای بردارهای واحد برای هر یک از ۳۳ حسگر مقداردهی اولیه می‌شوند.*
* ***محاسبه بردارهای واحد****:* 
  + *اسکریپت برای هر یک از ۳۳ حسگر تکرار می‌شود.*
  + *برای یک مدل کروی، بردار موقعیت یک حسگر (از مبدا تا حسگر) از قبل در آن نقطه بر سطح کره عمود است.*
  + *برای به دست آوردن بردار واحد، هر جزء از بردار موقعیت حسگر (xi​,yi​,zi​) بر اندازه بردار موقعیت تقسیم می‌شود. از آنجایی که تمام حسگرها روی یک کره با شعاع ثابت قرار دارند، اندازه به سادگی این شعاع (r3​=0.09 متر) است.* 
    - *exi​=xi​/radius*
    - *eyi​=yi​/radius*
    - *ezi​=zi​/radius*
* ***ذخیره بردارهای واحد****: اجزای بردار واحد محاسبه‌شده در Unit\_Vect\_coordinates.npz ذخیره می‌شوند.*
* ***نمایش****:* 
  + *یک نمودار سه‌بعدی تولید می‌شود.*
  + *از ax.quiver() برای رسم یک پیکان (نشان‌دهنده بردار واحد) در هر مکان حسگر استفاده می‌شود. پایه پیکان (xi​,yi​,zi​) و اجزای آن (exi​,eyi​,ezi​) هستند.*
  + *سطح نیمکره نیز رسم می‌شود.*

*توضیح خروجی:*

*اسکریپت یک نمایش سه‌بعدی (مشابه شکل ۴ در PDF) تولید می‌کند که نیمکره پوست سر را با یک بردار واحد جهت‌دار به بیرون (پیکان) در هر یک از ۳۳ مکان حسگر نشان می‌دهد. این بردارها در هر نقطه عمود بر سطح کروی به نظر می‌رسند.*

***د) مدل سر و ماتریس lead-field***

*سر با استفاده از سه کره همگن هم‌مرکز مدل‌سازی می‌شود:*

* *مغز (شعاع r1​=8.0 سانتی‌متر)*
* *جمجمه (شعاع r2​=8.5 سانتی‌متر)*
* *پوست سر (شعاع r3​=9.0 سانتی‌متر) هدایت‌ها:*
* *مغز و پوست سر: σ1​=σ3​=σ*
* *جمجمه: σ2​=σ/80*

*ماتریس lead-field MEG، G، منابع دوقطبی را به اندازه‌گیری‌های حسگر مرتبط می‌کند.*

***سوالات ماتریس lead-field (G)****:*

* ***ابعاد G****: ماتریس lead-field G ابعاد m×(3n) خواهد داشت، که m تعداد حسگرها و n تعداد مکان‌های دوقطبی است. در این مسئله، m=33 حسگر و n=105 مکان دوقطبی وجود دارد. هر دوقطبی دارای ۳ جزء متعامد است (مثلاً Qx، Qy، Qz که گشتاور آن را نشان می‌دهند). بنابراین، G 33×(3×105)=33×315 است.* 
  + *با این حال، توضیحات مسئله و کد اغلب با در نظر گرفتن یک بردار گشتاور دوقطبی q​ برای هر یک از n مکان، ساده‌سازی می‌کنند. اگر هر ستون j مربوط به یک مکان دوقطبی واحد باشد و ما میدان مغناطیسی ناشی از یک دوقطبی در آن مکان با جهت‌گیری خاص را محاسبه کنیم، ماتریس ممکن است به عنوان m×nd​ تفسیر شود، که nd​ تعداد کل اجزای دوقطبی مستقل در نظر گرفته شده است. اگر فرمول مورد استفاده، سهم یک منبع دوقطبی (بردار) در مکان j را در حسگر i محاسبه کند، آنگاه G m×n خواهد بود. صورت مسئله نشان می‌دهد که G "منابع دوقطبی" را به اندازه‌گیری‌ها مرتبط می‌کند. فعلاً فرض می‌کنیم، بر اساس فرمول‌بندی‌های معمول که در آن هر ستون نشان‌دهنده یک مکان منبع است و محاسبه، جهت‌گیری آن را در بر می‌گیرد، G m×nq​ است، که nq​ تعداد ستون‌هایی است که اثر اجزای هر دوقطبی را نشان می‌دهد (اغلب ۳ برای هر مکان).*
  + *با نگاهی به Task4.py و Task5.py، به نظر می‌رسد که j را از 0 تا n (تعداد دوقطبی‌ها، ۱۰۵) تکرار می‌کنند. Task5.py به طور خاص از یک دوقطبی واحد q=[0,0,1] استفاده می‌کند.*
  + *ماتریس lead-field G به طور موثر دارای m ردیف (تعداد حسگرها) و ND​ ستون است، که ND​ تعداد پارامترهای دوقطبی است. اگر هر یک از ۱۰۵ دوقطبی بتواند جهت‌گیری دلخواه داشته باشد (گشتاور با بردار سه‌بعدی qj​​=[qjx​,qjy​,qjz​] نشان داده می‌شود)، آنگاه G 33×(105×3)=33×315 خواهد بود.*
  + *با این حال، اگر ما میدان را برای یک دوقطبی واحد در یک جهت‌گیری خاص (مثلاً شعاعی یا مماسی) در هر یک از ۱۰۵ مکان محاسبه کنیم، یا اگر فرمول برای Gij​ میدان را در حسگر i به دلیل منبع j بدهد (که در آن منبع j یک کمیت برداری است)، تفسیر ممکن است کمی متفاوت باشد. PDF نشان می‌دهد که ستون ۷۵ نشان می‌دهد که چگونه یک دوقطبی خاص بر تمام ۳۳ حسگر تأثیر می‌گذارد. این نشان می‌دهد که هر ستون ممکن است مربوط به یک مکان منبع دوقطبی واحد باشد، و محاسبه Gij​ جهت‌گیری (گشتاور) مفروض دوقطبی را در بر می‌گیرد.*
  + *فرض می‌کنیم برای G در Task4.py، با توجه به ساختار حلقه، میدان را برای یک جهت‌گیری از پیش تعریف‌شده برای هر یک از ۱۰۵ دوقطبی محاسبه می‌کند. اگر منظور مسئله "یک مکان منبع دوقطبی خاص" باشد، و سپس جهت‌گیری تغییر کند، این پیچیدگی را اضافه می‌کند. با این حال، تعریف رایج Gij​ میدان در حسگر i به دلیل یک منبع دوقطبی واحد j است. اگر منبع j یک بردار باشد، آنگاه G ستون‌های بیشتری خواهد داشت.*
  + ***بازبینی****: عبارت PDF "رسم ستون ۷۵ (که نشان می‌دهد چگونه یک دوقطبی خاص بر تمام ۳۳ حسگر تأثیر می‌گذارد)" قویاً نشان می‌دهد که G m×n است (یعنی 33×105)، که در آن Gij​ میدان مغناطیسی شعاعی اسکالر در حسگر i به دلیل یک دوقطبی j از پیش تعریف‌شده (مکان و گشتاور) است.*
  + ***تفسیر نهایی برای ابعاد G****: G دارای ابعاد m×n است، که 33×105 می‌شود. هر عنصر Gij​ نشان‌دهنده قدرت میدان مغناطیسی شعاعی اندازه‌گیری‌شده توسط حسگر i به دلیل منبع دوقطبی واحد j (در مکان rqj​ با بردار گشتاور از پیش تعریف‌شده qj​) است.*
* ***معنای فیزیکی G[i,j]****: G[i,j] (یا Gij​) نشان‌دهنده میدان مغناطیسی (به طور خاص، جزء شعاعی آن اگر چیزی است که محاسبه می‌شود) اندازه‌گیری‌شده توسط حسگر i-ام است که توسط j-امین منبع دوقطبی تولید می‌شود، با فرض اینکه این دوقطبی دارای قدرت واحد و جهت‌گیری خاصی است. این حساسیت حسگر i به منبع j را کمی می‌کند.*
* ***چگونه G منابع دوقطبی را به اندازه‌گیری‌های حسگر مرتبط می‌کند****: ماتریس lead-field G به طور خطی اندازه‌ها (یا گشتاورهای) منبع دوقطبی را به اندازه‌گیری‌های حسگر مرتبط می‌کند. اگر S بردار قدرت‌ها/گشتاورهای منبع دوقطبی (n×1) و B بردار اندازه‌گیری‌های حسگر (m×1) باشد، آنگاه B=GS.*
* ***ستون ۷۵ از G****: ستون ۷۵ از G (یعنی G[:,74] اگر از نمایه‌گذاری مبتنی بر 0 برای ستون ۷۵ استفاده شود) قدرت میدان مغناطیسی اندازه‌گیری‌شده توسط هر یک از ۳۳ حسگر را در پاسخ به ۷۵امین منبع دوقطبی (با فرض قدرت واحد و جهت‌گیری از پیش تعریف‌شده) نشان می‌دهد. این یک بردار از ۳۳ مقدار است که الگوی فضایی میدان مغناطیسی را در تمام حسگرها به دلیل آن دوقطبی خاص نشان می‌دهد.*

***تکلیف ۴: محاسبه ماتریس lead-field (G)***

***هدف****: پیاده‌سازی محاسبه ماتریس lead-field با استفاده از task4.py و رسم ستون ۷۵.*

***توضیح کد (****task4.py****)****:*

* ***بارگذاری داده‌ها****:* 
  + *مختصات برای دوقطبی‌ها (Dipole\_coordinates.npz)، حسگرها (sensor\_coordinates.npz)، و بردارهای واحد (Unit\_Vect\_coordinates.npz) بارگذاری می‌شوند.*
  + *این‌ها در آرایه‌هایی سازماندهی می‌شوند: rq (موقعیت‌های دوقطبی)، r (موقعیت‌های حسگر)، er (بردارهای واحد حسگر).*
* ***ثابت‌ها****: پارامترهای مدل سر (m,n,R0,R1,R2,R3,sigma,μ) مطابق PDF تعریف می‌شوند.*
* ***مقداردهی اولیه G****: ماتریس lead-field G به عنوان یک آرایه NumPy از صفرها با ابعاد m×n (33×105) مقداردهی اولیه می‌شود.*
* ***محاسبه اجزای G****:* 
  + *کد برای هر حسگر i (از 0 تا m−1) و هر دوقطبی j (از 0 تا n−1) تکرار می‌شود.*
  + *درون حلقه، G[i,j] محاسبه می‌شود. فرمول میدان مغناطیسی تولیدشده توسط یک دوقطبی جریانی در یک هادی کروی استفاده می‌شود. برای MEG، ما معمولاً به جزء شعاعی میدان مغناطیسی خارج از کره علاقه‌مند هستیم.*
  + *فرمول Sarvas اغلب برای یک مدل کروی استفاده می‌شود: B(r)=4πμ0​​F2Fq​×rq​−(q​×rq​⋅r)∇F​ که rq​ مکان دوقطبی، q​ گشتاور دوقطبی، r مکان حسگر است. a=∣r−rq​∣، F=a(∣r∣∣r−rq​∣+∣r∣2−rq​⋅r)، و ∇F=(∣r−rq​∣/∣r∣+(r⋅a)/a3+2)r−(a/∣r∣+2+(r⋅a)/a3)rq​.*
  + *جزء شعاعی Br​=B(r)⋅er​ است، که er​=r/∣r∣ بردار واحد برای جهت‌گیری حسگر است.*
  + *utility\_functions.py برای MEG تابع Calc\_G(er, r, rq) را ارائه می‌دهد. جای خالی در task4.py شامل فراخوانی این تابع خواهد بود. فرض می‌کنیم Calc\_G بردار Bij​ (بردار میدان مغناطیسی در حسگر i به دلیل دوقطبی j با یک گشتاور ثابت، مثلاً گشتاور واحد در جهت z برای تمام دوقطبی‌ها به عنوان یک ساده‌سازی برای ساخت G) را محاسبه می‌کند. سپس G[i,j] برابر با Bij​⋅eri​ خواهد بود.*
  + *اگر q (بردار گشتاور دوقطبی برای منبع j) باید برای هر دوقطبی تعریف شود (مثلاً به عنوان [0,0,1] برای همه، یا متغیر)، این باید انجام شود. برای تکلیف ۴، معمول است که هنگام ساخت G، یک جهت‌گیری کانونی برای هر دوقطبی فرض شود. فرض می‌کنیم برای هر دوقطبی j، یک بردار گشتاور دوقطبی qj​ (مثلاً [0,0,1] برای سادگی، یا بر اساس جهت‌گیری خودش اگر از پیش تعریف‌شده باشد) استفاده می‌شود.*
  + *خط G[i, j] = Calc\_G(er[i], r[i], rq[j], q[j]) (با فرض اینکه Calc\_G q را می‌گیرد و جزء شعاعی را برمی‌گرداند). یا به طور صریح‌تر: R = r[i] - rq[j] (بردار از دوقطبی به حسگر) R\_norm = np.linalg.norm(R) G[i,j] = (mu / (4 \* np.pi \* R\_norm\*\*3)) \* np.dot(np.cross(q[j], rq[j]), er[i]) (این یک فرمول ساده‌شده برای دوقطبی دور از مبدا است، یک فرمول کامل‌تر مانند Sarvas باید استفاده شود، یا فرمولی که توسط utility\_functions.Calc\_G پیشنهاد می‌شود). utility\_functions.Calc\_G(er, r, rq) ارائه‌شده برمی‌گرداند (mu / (4 \* np.pi)) \* (np.cross(er, rq)) / (np.linalg.norm(r - rq)). به نظر می‌رسد این فاقد گشتاور دوقطبی q است و احتمالاً بخشی میانی از یک فرمول یا برای یک مقدار مستقیم lead-field نادرست است. یک فرمول ساده‌شده رایج برای میدان مغناطیسی از یک دوقطبی جریانی Q​ در r′ اندازه‌گیری‌شده در r عبارت است از B(r)=4πμ0​​∣r−r′∣3Q​×(r−r′)​. جزء شعاعی در حسگر i برابر با Br,i​=B(ri​​)⋅er,i​ خواهد بود. فرض می‌کنیم q\_vectors یک آرایه n×3 از گشتاورهای دوقطبی است. سپس برای هر j: Q\_j = q\_vectors[j] B\_vec = (mu / (4 \* np.pi)) \* (np.cross(Q\_j, r[i] - rq[j])) / (np.linalg.norm(r[i] - rq[j])\*\*3) G[i,j] = np.dot(B\_vec, er[i])*
* ***نمایش****: ستون ۷۵ از G (یعنی G[:, 74]) به عنوان یک نمودار خطی رسم می‌شود، با شاخص حسگر روی محور x و مقدار lead-field روی محور y.*

*توضیح خروجی (task4.py):*

*اسکریپت یک نمودار خروجی می‌دهد که مقادیر ستون ۷۵ ماتریس lead-field G را نشان می‌دهد. این نمودار ۳۳ نقطه خواهد داشت که مربوط به قدرت میدان مغناطیسی است که هر حسگر به دلیل ۷۵امین منبع دوقطبی اندازه‌گیری می‌کند (با فرض یک جهت‌گیری و قدرت کانونی برای این دوقطبی). شکل این نمودار نشان می‌دهد که چگونه تأثیر آن دوقطبی خاص در آرایه حسگر پخش می‌شود.*

*تأثیر تغییر نسبت هدایت:*

*lead-field MEG برای یک هادی با تقارن کروی از نظر تئوری مستقل از توزیع هدایت شعاعی است. این بدان معناست که تغییر نسبت هدایت بین بافت جمجمه و مغز (مثلاً σ2​=σ/80 در مقابل σ2​=σ/100) نباید ماتریس lead-field MEG، G، را تغییر دهد، تا زمانی که مدل متقارن کروی باقی بماند و هدایت‌ها در هر پوسته کروی همگن باشند. این یک مزیت کلیدی MEG نسبت به EEG است. میدان‌های مغناطیسی نسبتاً بدون مزاحمت از این بافت‌ها عبور می‌کنند.*

*مدل واقعی سر در مقابل مدل کروی:*

*اگر ما به جای یک مدل کروی از یک مدل واقعی سر (مثلاً مشتق‌شده از MRI، با شکل دقیق مغز، جمجمه و پوست سر) استفاده می‌کردیم:*

* *ماتریس lead-field G به طور قابل توجهی تغییر می‌کرد. فرض تقارن کروی دیگر برقرار نبود.*
* *محاسبات بسیار پیچیده‌تر می‌شدند و معمولاً به روش‌های عددی مانند روش اجزای مرزی (BEM) یا روش اجزای محدود (FEM) نیاز داشتند.*
* *استقلال MEG از تغییرات هدایت شعاعی کمتر دقیق بود، زیرا جریان‌ها ممکن است توسط مرزهای غیر کروی منحرف یا هدایت شوند، اگرچه MEG هنوز کمتر از EEG به عدم قطعیت‌های هدایت حساس است.*
* *دقت مکان‌یابی منبع می‌تواند بهبود یابد اگر مدل واقعی، هندسه واقعی سر و جریان جریان را بهتر نشان دهد.*

***تکلیف ۵: میدان‌های مغناطیسی شعاعی از یک منبع دوقطبی واحد***

***هدف****: محاسبه و نمایش جزء شعاعی چگالی شار مغناطیسی (Br) در هر حسگر MEG به دلیل یک منبع دوقطبی واحد. دوقطبی در (ro​,θo​,ϕo​)=(7cm,45∘,45∘) با گشتاور q​0​=[0,0,1] (جهت‌دار در امتداد محور z) قرار دارد. از داده‌های حسگر sensor\_coordinates.npz استفاده کنید.*

***توضیح کد (****task5.py****)****:*

* ***بارگذاری داده‌ها****: مختصات حسگر (r\_x, r\_y, r\_z, r) و بردارهای واحد (er\_x, er\_y, er\_z, er) بارگذاری می‌شوند.*
* ***تعریف پارامترهای دوقطبی****:* 
  + *مختصات کروی برای دوقطبی: $\text{theta\_dipole} = 45^{\circ}$، $\text{phi\_dipole} = 45^{\circ}$، $\text{radius\_dipole} = 0.07$ متر.*
  + *این‌ها با استفاده از Conv\_coordinates به مختصات دکارتی rq = [x\_dipole, y\_dipole, z\_dipole] تبدیل می‌شوند.*
  + *بردار گشتاور دوقطبی q = np.array([0, 0, 1]) تعریف می‌شود.*
* ***ثابت‌ها****: ثابت‌های استاندارد (m=33,n=1,μ و غیره) تعریف می‌شوند.*
* ***محاسبه Br****:* 
  + *اسکریپت چگالی شار مغناطیسی شعاعی B\_r را در هر یک از m=33 حسگر به دلیل این دوقطبی واحد محاسبه می‌کند.*
  + *یک حلقه برای هر حسگر i تکرار می‌شود.*
  + *درون حلقه، بردار میدان مغناطیسی Bi​ در حسگر i به دلیل دوقطبی q​ در rq​ محاسبه می‌شود. با استفاده از قانون بیو-ساوار برای یک دوقطبی جریانی یا فرمول Sarvas: B(ri​​)=4πμ0​​∣ri​​−rq​​∣3q​×(ri​​−rq​​)​ (این یک فرم ساده‌شده است که اغلب استفاده می‌شود؛ یک فرمول کامل‌تر مانند Sarvas برای مدل‌های کروی بهتر است اما این برای иллюстрация رایج است).*
  + *سپس جزء شعاعی Br,i​=B(ri​​)⋅er,i​ است، که er,i​ بردار واحد برای حسگر i است.*
  + *B\_r یک آرایه از ۳۳ مقدار خواهد بود.*
  + *به طور متناوب، اگر ماتریس G از تکلیف ۴ برای دوقطبی‌هایی با گشتاور [0,0,1] بود، آنگاه B\_r می‌توانست با انتخاب ستون G مربوط به دوقطبی نزدیک‌ترین به (7cm,45∘,45∘) و ضرب آن در گشتاور واقعی آن (اگر G برای گشتاورهای واحد بود) به دست آید. با این حال، تکلیف ۵ یک محاسبه مستقیم را می‌خواهد.*
  + *TODO: Initialize the lead field matrix G و حلقه برای محاسبه G ممکن است کمی از عنوان PDF "میدان‌های مغناطیسی شعاعی از یک منبع دوقطبی واحد" منحرف شده باشد. هسته کار محاسبه B\_r برای ۳۳ حسگر از این یک دوقطبی است. اگر G در اینجا محاسبه شود، یک ماتریس 33×1 خواهد بود. B\_r = G @ q بعدی (اگر G به طور مناسب تعریف شده بود) یا B\_r = G[:,0] اگر G 33×1 باشد و q گنجانده شده باشد.*
  + *ساختار ارائه‌شده Task5.py نشان می‌دهد که ابتدا یک ماتریس G محاسبه شود، حتی برای n=1. بنابراین، G 33×1 خواهد بود. سپس B\_r از G مشتق می‌شود. به عنوان مثال، G[i, 0] میدان از این دوقطبی واحد (j=0) در حسگر i خواهد بود. سپس B\_r = G[:,0].*
* ***ذخیره G****: ماتریس G محاسبه‌شده 33×1 در MEG\_Laed\_Field.npz ذخیره می‌شود.*
* ***نمایش****:* 
  + *یک نمودار پراکندگی سه‌بعدی از حسگرها، مشابه شکل ۱، ایجاد می‌شود.*
  + *رنگ هر نقطه حسگر با مقدار B\_r محاسبه‌شده در آن حسگر، با استفاده از یک نقشه رنگی (مثلاً 'hot') تعیین می‌شود. یک نوار رنگی نگاشت از رنگ به اندازه B\_r را نشان می‌دهد.*
  + *مکان دوقطبی و جهت‌گیری آن (بردار گشتاور) با استفاده از ax.quiver رسم می‌شود.*
  + *سطح نیمکره نشان داده می‌شود.*

*توضیح خروجی (task5.py):*

*اسکریپت یک نمایش سه‌بعدی (مشابه شکل ۶ در PDF) خروجی می‌دهد. ۳۳ حسگر MEG روی سطح پوست سر نمایش داده می‌شوند. هر حسگر بر اساس قدرت میدان مغناطیسی شعاعی (Br​) که از دوقطبی واحد مشخص‌شده تشخیص می‌دهد، رنگ‌آمیزی می‌شود. یک بردار قرمز مکان و جهت‌گیری دوقطبی منبع را نشان می‌دهد. این نمودار توزیع فضایی میدان مغناطیسی تولیدشده توسط آن یک دوقطبی را نشان می‌دهد.*

***پاسخ به سوالات تکلیف ۵****:*

* ***جزء شعاعی چگالی شار مغناطیسی (Br) و اهمیت آن****:* 
  + *Br​ جزء بردار میدان مغناطیسی است که عمود (شعاعی) بر سطح سر (کروی مفروض) در مکان حسگر است.*
  + *این در MEG، به ویژه با مدل‌های سر متقارن کروی، مهم است، زیرا MEG عمدتاً جزء میدان مغناطیسی عمود بر جمجمه را اندازه‌گیری می‌کند. برای دوقطبی‌های شعاعی در یک هادی متقارن کروی، میدان مغناطیسی خارجی صفر است. MEG بیشتر به دوقطبی‌های مماسی حساس است که میدان‌های مغناطیسی شعاعی تولید می‌کنند. سیستم‌های مدرن ممکن است از گرادیومترها یا مگنتومترهای برداری استفاده کنند که می‌توانند اجزای دیگر را حس کنند، اما جزء شعاعی اساسی است.*
* ***فاصله بین حسگر و منبع دوقطبی در مقابل قدرت میدان مغناطیسی****:* 
  + *قدرت میدان مغناطیسی با فاصله از منبع دوقطبی به سرعت کاهش می‌یابد. برای یک دوقطبی، قدرت میدان معمولاً با 1/r2 یا 1/r3 کاهش می‌یابد، بسته به پیکربندی خاص (مثلاً تقریب میدان دور 1/r2، اجزای میدان نزدیک می‌توانند 1/r3 داشته باشند). بنابراین، حسگرهای نزدیک‌تر به منبع دوقطبی، میدان مغناطیسی به طور قابل توجهی قوی‌تری را اندازه‌گیری می‌کنند.*
* ***قوی‌ترین اندازه‌گیری‌های میدان مغناطیسی و چرا****:* 
  + *بر اساس نمایش، قوی‌ترین اندازه‌گیری‌های میدان مغناطیسی (نشان داده شده با رنگ، مثلاً زرد/سفید روشن اگر نقشه رنگی 'hot' برای مقادیر مثبت بزرگ استفاده شود، و قرمز تیره/سیاه برای مقادیر منفی بزرگ) در حسگرهای نزدیک‌ترین به منبع دوقطبی مشاهده می‌شود، اما همچنین به جهت‌گیری دوقطبی بستگی دارد.*
  + *برای یک دوقطبی جهت‌دار در امتداد محور z (گشتاور [0,0,1]) در (7cm,45∘,45∘)، خطوط میدان مغناطیسی به دور دوقطبی حلقه می‌زنند. جزء شعاعی Br​ در نواحی بالا و کناره‌های تصویر مماسی دوقطبی روی کره حسگر، قوی‌ترین (هم قله‌های مثبت و هم منفی) خواهد بود و یک الگوی دو لوب مشخص (یکی مثبت، یکی منفی) را تشکیل می‌دهد. آن‌ها در این مکان‌ها رخ می‌دهند زیرا این حسگرها به طور بهینه برای گرفتن خطوط شار میدان مغناطیسی هنگامی که به صورت شعاعی از پوست سر خارج یا دوباره وارد می‌شوند، قرار گرفته و جهت‌گیری شده‌اند. حسگرهایی که مستقیماً "بالای" یک منبع کاملاً مماسی در یک مدل کروی قرار دارند، قوی‌ترین میدان‌های شعاعی را اندازه‌گیری می‌کنند.*

***تکلیف ۶: شبیه‌سازی یک سیگنال MEG متغیر با زمان از یک منبع دوقطبی***

*هدف: شبیه‌سازی یک سیگنال MEG متغیر با زمان در حسگر شماره ۳۰ از یک دوقطبی با موقعیت ثابت (۷ سانتی‌متر، 45∘,45∘) و جهت‌گیری [0,0,1]، اما با اندازه متغیر با زمان w(t).*

*تابع عبارت است از w(t)=120sin(8πt)+45sin(14πt)+30sin(20πt)+15sin(40πt)+5sin(80πt).*

*۱ ثانیه داده با فرکانس نمونه‌برداری ۱۰۰۰ هرتز تولید کنید.*

***توضیح کد (****task6.py****)****:*

* ***تعریف*** *w(t):* 
  + *یک تابع پایتون w\_func(t) تعریف می‌شود که تابع متناوب داده‌شده w(t) را پیاده‌سازی می‌کند.*
* ***ایجاد بردار زمان****:* 
  + *یک بردار زمان t با استفاده از np.linspace(0, 1, 1000) برای ۱ ثانیه داده با ۱۰۰۰ نمونه (۱۰۰۰ هرتز) ایجاد می‌شود. شکل PDF ۲ ثانیه را نشان می‌دهد، بنابراین اگر هدف تطابق با شکل باشد، np.linspace(0, 2, 2000) ممکن است مورد نظر باشد، اما متن ۱ ثانیه را می‌گوید. اجازه دهید مطابق متن به ۱ ثانیه پایبند باشیم. نمودارهای شکل ۷، ۲ ثانیه را نشان می‌دهند. من برای سازگاری با نمودار، ۲ ثانیه را برای t فرض می‌کنم. بنابراین، t = np.linspace(0, 2, 2000, endpoint=False).*
  + *w\_values با اعمال w\_func به t محاسبه می‌شوند.*
* ***بارگذاری ماتریس lead-field / محاسبه میدان ثابت****:* 
  + *مسئله به محاسبه سیگنال در حسگر شماره ۳۰ اشاره دارد. ما به مقدار lead-field G[29,j0​] (نمایه‌گذاری مبتنی بر 0 برای حسگر ۳۰) برای دوقطبی خاص j0​ (در ۷ سانتی‌متر، 45∘,45∘ با جهت‌گیری [0,0,1]) نیاز داریم.*
  + *این مقدار lead-field G30,dipole​ نشان‌دهنده میدان مغناطیسی ثابت در حسگر شماره ۳۰ به دلیل یک دوقطبی با قدرت واحد با جهت‌گیری داده‌شده است. این می‌تواند از MEG\_Laed\_Field.npz تولیدشده در تکلیف ۵ گرفته شود (این ردیف ۳۰ام ماتریس 33×1 G خواهد بود، یعنی G[29,0] اگر حسگر ۳۰ شاخص ۲۹ باشد). اجازه دهید این Br\_static\_unit​ باشد.*
  + *بردار q (جهت‌گیری دوقطبی [0,0,1]) تعریف می‌شود. lead-field Gi,j​ (از تکلیف ۵ برای این دوقطبی واحد) از قبل جهت‌گیری و موقعیت آن را برای دادن Br​ در حسگر i برای یک گشتاور واحد در بر می‌گیرد. بنابراین، اگر Gtask5​[29,0] Br​ برای q=[0,0,1] (گشتاور واحد) باشد، آنگاه این Br\_static​ است. عبارت مسئله در اینجا کمی مبهم است.*
  + *فرض می‌کنیم Gsensor30,dipole​ مقدار lead-field است: میدان مغناطیسی شعاعی در حسگر ۳۰ اگر دوقطبی دارای اندازه واحد بود.*
  + *B\_r (میدان ثابت): این Gsensor30,dipole​×1 است (با فرض اینکه w(t) یک ضریب مقیاس‌بندی است که به یک گشتاور دوقطبی واحد اعمال می‌شود). نمودار یک مقدار ثابت را نشان می‌دهد، که نشان می‌دهد این میدان در صورتی است که w(t) به طور مداوم ۱ بود.*
* ***محاسبه میدان متغیر با زمان****:* 
  + *B\_r\_new = G\_sensor30\_dipole \* w\_values. این مقدار lead-field ایستا را با اندازه متغیر با زمان w(t) مقیاس‌بندی می‌کند.*
* ***نمایش****:* 
  + *سه زیرنمودار (مشابه شکل ۷ در PDF) ایجاد می‌شوند:* 
    1. *w(t) در مقابل زمان.*
    2. *میدان مغناطیسی ثابت Br​ (محاسبه‌شده گویی w(t)=1 یا مقداری مرجع) در حسگر شماره ۳۰ در مقابل زمان.*
    3. *میدان مغناطیسی متغیر با زمان Br​\_new در حسگر شماره ۳۰ در مقابل زمان.*

*توضیح خروجی (task6.py):*

*اسکریپت یک شکل با سه نمودار همانطور که در بالا توضیح داده شد، تولید می‌کند.*

1. *نمودار اول شکل موج متناوب پیچیده w(t) را نشان می‌دهد.*
2. *نمودار دوم یک خط صاف را نشان می‌دهد که نشان‌دهنده میدان مغناطیسی ثابتی است که حسگر شماره ۳۰ در صورتی که اندازه دوقطبی ثابت بود (مثلاً با قدرت واحد) اندازه‌گیری می‌کرد.*
3. *نمودار سوم سیگنال MEG شبیه‌سازی‌شده در حسگر شماره ۳۰ را نشان می‌دهد. شکل آن w(t) را منعکس می‌کند، که با مقدار lead-field که آن دوقطبی را به حسگر شماره ۳۰ متصل می‌کند، مقیاس‌بندی شده است.*

*رابطه بین w(t) و Br​\_new:*

*رابطه خطی است. Br​\_new(t)=G30,dipole​⋅w(t). میدان مغناطیسی حاصل Br​\_new در حسگر شماره ۳۰ مستقیماً با اندازه متغیر با زمان دوقطبی w(t) متناسب است. شکل شکل موج Br​\_new(t) با شکل موج w(t) یکسان خواهد بود، فقط با ثابت G30,dipole​ (که نشان می‌دهد حسگر شماره ۳۰ چقدر قوی آن دوقطبی خاص را "می‌بیند") مقیاس‌بندی شده است.*

*شبیه‌سازی دو دوقطبی با الگوهای متغیر با زمان متفاوت:*

*برای شبیه‌سازی دو دوقطبی با الگوهای متغیر با زمان متفاوت، w1​(t) و w2​(t):*

1. *دو تابع متغیر با زمان، w1\_values و w2\_values را تعریف کنید.*
2. *مقادیر lead-field را برای هر دوقطبی در حسگر شماره ۳۰ تعیین کنید: G30,dipole1​ و G30,dipole2​. این مقادیر به موقعیت و جهت‌گیری هر یک از دو دوقطبی نسبت به حسگر شماره ۳۰ بستگی دارد.*
3. *سیگنال را از هر دوقطبی به طور مستقل محاسبه کنید:* 
   * *Br\_new1​(t)=G30,dipole1​⋅w1\_values(t)*
   * *Br\_new2​(t)=G30,dipole2​⋅w2\_values(t)*
4. *سیگنال کل اندازه‌گیری‌شده در حسگر شماره ۳۰، مجموع سیگنال‌های هر دوقطبی به دلیل اصل برهم‌نهی خواهد بود:* 
   * *Br\_total​(t)=Br\_new1​(t)+Br\_new2​(t)*
5. *کد نیاز به موارد زیر دارد:* 
   * *تعریف موقعیت‌ها و جهت‌گیری‌ها برای دو دوقطبی.*
   * *محاسبه یا بارگذاری مقادیر lead-field مربوطه آن‌ها به حسگر شماره ۳۰.*
   * *تولید دو سری w(t) متفاوت.*
   * *محاسبه دو سری B\_r\_new و جمع آن‌ها.*

*تغییر سیگنال MEG اگر جهت‌گیری دوقطبی در طول زمان تغییر کند:*

*اگر جهت‌گیری دوقطبی q​(t) به جای فقط اندازه‌اش در طول زمان تغییر می‌کرد:*

* *lead-field Gi,j​ خود معمولاً برای یک جهت‌گیری منبع ایستا تعریف می‌شود. اگر جهت‌گیری q​j​(t) تغییر کند، آنگاه lead-field موثر وابسته به زمان می‌شود، یا به طور دقیق‌تر، بردار منبع q​j​(t) وابسته به زمان است.*
* *محاسبه Br,i​(t)=function(ri​,er,i​,rq,j​,q​j​(t)) خواهد بود.*
* *به عنوان مثال، اگر q​j​(t)=[qx​(t),qy​(t),qz​(t)]:* 
  + *میدان مغناطیسی در حسگر i برابر با Bi​(t)=Lix​qx​(t)+Liy​qy​(t)+Liz​qz​(t) خواهد بود، که Lik​ بردار lead-field برای یک دوقطبی واحد در جهت k در منبع j است. جزء شعاعی Br,i​(t)=(Lix​⋅er,i​)qx​(t)+(Liy​⋅er,i​)qy​(t)+(Liz​⋅er,i​)qz​(t) خواهد بود.*
* *سیگنال MEG حاصل Br​\_new(t) پیچیده‌تر خواهد بود. این فقط یک نسخه مقیاس‌بندی‌شده از یک w(t) واحد نخواهد بود.*
* ***آزمایش****:* 
  + *اجزای متغیر با زمان را برای گشتاور دوقطبی تعریف کنید، مثلاً qx​(t),qy​(t),qz​(t). به عنوان مثال، q​(t)=[sin(ω1​t),cos(ω2​t),constant]⋅w(t).*
  + *در هر نقطه زمانی، بردار میدان مغناطیسی B(t) تولیدشده توسط q​(t) را در مکان حسگر شماره ۳۰ محاسبه کنید.*
  + *این B(t) را روی بردار شعاعی واحد حسگر شماره ۳۰، er,30​، تصویر کنید تا Br,30​(t) به دست آید.*
  + *شکل موج Br,30​(t) احتمالاً به طور قابل توجهی با w(t) متفاوت خواهد بود زیرا "بهره" (تصویر میدان روی جهت حساس حسگر) نیز با زمان به دلیل چرخش گشتاور دوقطبی تغییر می‌کند. این می‌تواند اجزای فرکانسی جدیدی را معرفی کند یا اجزای موجود از w(t) را مدوله کند.*

***۲. مسئله مستقیم EEG***

***تکلیف ۷: فرمول پتانسیل سطحی (دوقطبی در [0,0,z])***

***هدف****: برای یک منبع جریان الکتریکی m=[mx​,my​,mz​] در [0,0,z0​] (توجه: PDF می‌گوید [0,0,z] ، من از z0​ برای مختصات z دوقطبی استفاده می‌کنم تا با مختصات z حسگر اشتباه نشود) در مدل کروی سه‌لایه، رابطه ریاضی پتانسیل الکتریکی V را در یک حسگر EEG بر حسب m به صورت ماتریسی-اسکالر بنویسید.*

*رابطه ریاضی:*

*پتانسیل الکتریکی V در مکان حسگر r (با مختصات کروی (r,θ,ϕ)، که r=R3​ شعاع پوست سر است) به دلیل یک دوقطبی جریانی m در r0​=(0,0,z0​) می‌تواند با استفاده از یک مجموع روی چندجمله‌ای‌های لژاندر بیان شود.*

*برای یک دوقطبی m در r0​، پتانسیل V(r) به این صورت داده می‌شود:*

*V(r)=4πσ1​1​∑n=1∞​(2n+1)(∣r∣∣r0​∣​)n+1∣r0​∣dn​​(∣r0​∣m⋅r0​​Pn​(cosγ)−(m⋅u^n​)n(n+1)Pn1​(cosγ)​)*

*این پیچیده است. به طور کلی‌تر، V(r)=L(r,r0​)⋅m، که L یک بردار lead-field است.*

*پتانسیل V در موقعیت حسگر rs​ (روی سطح پوست سر، شعاع R3​) به دلیل یک دوقطبی m=[mx​,my​,mz​] در r0​=(0,0,z0​) می‌تواند به این صورت نوشته شود:*

*V(rs​)=Lx​(rs​,r0​)mx​+Ly​(rs​,r0​)my​+Lz​(rs​,r0​)mz​.*

*این از قبل به صورت ضرب داخلی است: V(rs​)=Lrs​,r0​​⋅m.*

*که Lrs​,r0​​=[Lx​(rs​,r0​),Ly​(rs​,r0​),Lz​(rs​,r0​)] بردار lead-field برای EEG در حسگر rs​ برای یک دوقطبی در r0​ است.*

*فرمول معمولاً شامل عباراتی مانند این است:*

*V=4πσ3​R3​1​∑n=1∞​(n+1)(2n+1)2dn​​(R3​R0​​)nPn​(cosγ)(m⋅R^0​)+… (عبارات برای اجزای مماسی)*

*که σ3​ هدایت پوست سر، R0​=∣r0​∣، γ زاویه بین rs​ و r0​ است. Pn​ چندجمله‌ای‌های لژاندر هستند. dn​ ضرایبی هستند که به شعاع‌ها و هدایت‌های کره‌ها بستگی دارند.*

*فرض کنید rs​=(R3​,θs​,ϕs​) مکان حسگر و r0​=(z0​,0,0) در سیستم کروی خودش (هم‌راستا با محور z) باشد.*

*زاویه γ بین rs​ (حسگر) و r0​ (دوقطبی در (0,0,z0​)) به سادگی زاویه قطبی θs​ حسگر است.*

*گشتاور دوقطبی m=mr​r^0​+mθ​θ^0​+mϕ​ϕ^​0​.*

*اگر m=[mx​,my​,mz​]:*

*برای یک دوقطبی در (0,0,z0​)، جهت شعاعی از مبدا به دوقطبی k^ (محور z) است.*

*mr​=mz​.*

*اجزای مماسی mx​,my​ در صفحه xy هستند.*

*پتانسیل V در حسگر (R3​,θ,ϕ) می‌تواند به این صورت نوشته شود:*

*V(R3​,θ,ϕ)=4πσN​1​∑n=1∞​(R3​r0​​)n−1R32​dn′​​[n(m⋅r^0​)Pn​(cosθ)+(m⋅θ^0​)cos(ϕ−ϕ0​)Pn1​(cosθ)+(m⋅ϕ^​0​)sin(ϕ−ϕ0​)Pn1​(cosθ)]*

*(این یک فرم کلی است؛ برای r0​ روی محور z، که ϕ0​ برای جزء شعاعی بی‌ربط است و θ0​=0، ساده‌سازی رخ می‌دهد).*

*برای r0​=(0,0,z0​):*

*دوقطبی می‌تواند به یک جزء شعاعی mz​ (در امتداد محور z) و اجزای مماسی mx​,my​ (در صفحه xy) تجزیه شود.*

*V=Vradial​+Vtangential​*

*Vradial​(rs​;mz​k^)=4πσs​Rs​mz​​∑n​(2n+1)(Rs​z0​​)n−1ngn​​Pn​(cosθs​)*

*Vtangential​(rs​;mx​i^+my​j^​)=4πσs​Rs​1​∑n​(2n+1)(Rs​z0​​)n−1nhn​​(mx​cosϕs​+my​sinϕs​)Pn1​(cosθs​)*

*که gn​,hn​ ضرایب از مدل سه‌پوسته، σs​ هدایت پوست سر، Rs​ شعاع پوست سر است.*

*به صورت ماتریسی-اسکالر برای حسگر i و دوقطبی j:*

*Vi​=[Lix(j)​Liy(j)​Liz(j)​]​mjx​mjy​mjz​​​=Li(j)​⋅mj​*

*هر Lik(j)​ تابعی از موقعیت حسگر i (R3​,θi​,ϕi​) و موقعیت دوقطبی j (0,0,z0j​) است.*

***تکلیف ۸: مکان دوقطبی دلخواه برای EEG***

***هدف****: رابطه تکلیف ۷ را در صورتی که دوقطبی در یک نقطه دلخواه (r0​,θ0​,ϕ0​) باشد، اصلاح کنید. بحث یا اثبات کنید که آیا جایگزینی ϕ با (ϕ−ϕ0​) و θ با (θ−θ0​) در فرمول کافی است یا خیر.*

*اگر دوقطبی در یک مکان دلخواه r0​=(r0​,θ0​,ϕ0​) باشد، عبارت پتانسیل عمدتاً به این دلیل پیچیده‌تر می‌شود که زاویه γ بین موقعیت حسگر rs​=(R3​,θs​,ϕs​) و موقعیت دوقطبی r0​ دیگر به سادگی θs​ نیست.*

*با قانون کسینوس‌های کروی:*

*cosγ=cosθs​cosθ0​+sinθs​sinθ0​cos(ϕs​−ϕ0​).*

*چندجمله‌ای‌های لژاندر Pn​(cosγ) و توابع وابسته لژاندر Pnk​(cosγ) در بسط سری، اکنون به این cosγ پیچیده‌تر بستگی خواهند داشت.*

*این ادعا که می‌توان به سادگی ϕ را با (ϕs​−ϕ0​) و θ را با (θs​−θ0​) جایگزین کرد،* ***به طور کلی برای آرگومان خود چندجمله‌ای‌های لژاندر اگر θ0​=0 باشد، کافی یا صحیح نیست****.*

* *عبارت (ϕs​−ϕ0​) به درستی اختلاف سمتی را نشان می‌دهد و در بسط شامل Pnk​(cosγ)cos(k(ϕs​−ϕ0′​)) استفاده می‌شود که ϕ0′​ زاویه سمتی تصویر rs​−r0​ در یک سیستم مختصات خاص است.*
* *با این حال، جایگزینی θ با (θs​−θ0​) در Pn​(cosθ) نادرست است. آرگومان باید cosγ باشد.*

*قضیه جمع برای هارمونیک‌های کروی معمولاً برای چنین مواردی فراخوانی می‌شود:*

*Pn​(cosγ)=∑m=−nn​Yn,m∗​(θ0​,ϕ0​)Yn,m​(θs​,ϕs​)، که Yn,m​ هارمونیک‌های کروی هستند.*

*یا، Pn​(cosγ)=Pn​(cosθs​)Pn​(cosθ0​)+2∑k=1n​(n+k)!(n−k)!​Pnk​(cosθs​)Pnk​(cosθ0​)cos(k(ϕs​−ϕ0​)).*

*بنابراین، اصلاح شامل جایگزینی Pn​(cosθs​) (از حالت دوقطبی-روی-محور-z) با Pn​(cosγ) (یا بسط آن) است. عبارات برای اجزای مماسی نیز به دلیل جهت‌گیری m نسبت به r0​ و rs​ پیچیده‌تر می‌شوند.*

*جایگزینی زاویه ϕ که در عباراتی مانند cos(mϕ) یا sin(mϕ) (مربوط به زاویه سمتی حسگر در سیستم مختصات محلی دوقطبی) ظاهر می‌شود با (ϕs​−ϕ0​) برای در نظر گرفتن موقعیت سمتی دوقطبی کافی است. زاویه قطبی θs​ (ارتفاع حسگر) و θ0​ (ارتفاع دوقطبی) با این اختلاف سمتی ترکیب می‌شوند تا cosγ را تشکیل دهند. جایگزینی ساده (θs​−θ0​) برای θ صحیح نیست.*

***تکلیف ۹: ماتریس lead-field EEG (L)***

***هدف****: محاسبه ماتریس lead-field EEG (L) با استفاده از Task9.py، که منابع دوقطبی را به اندازه‌گیری‌های پتانسیل الکتریکی مرتبط می‌کند. رسم ستون ۷۵.*

***توضیح کد (****Task9.py****)****:*

* ***بارگذاری داده‌ها****: مختصات دوقطبی (rq\_x, rq\_y, rq\_z) و مختصات حسگر (r\_x, r\_y, r\_z) از فایل‌های .npz آن‌ها بارگذاری می‌شوند.*
* ***تبدیل به کروی****: هم مختصات دکارتی دوقطبی و هم حسگر با استفاده از یک تابع cartesian\_to\_polar (احتمالاً در utility\_functions.py) به مختصات کروی (r\_0, theta\_0, phi\_0 برای دوقطبی‌ها؛ r, theta, phi برای حسگرها) تبدیل می‌شوند.*
* ***ثابت‌ها****: پارامترهای مدل سر (m=33,n=105,R0,R1,R2,R3,sigma و غیره) تعریف می‌شوند. σ1​,σ2​,σ3​ بر اساس sigma و نسبت σ/80 برای جمجمه هستند.*
* ***مقداردهی اولیه L****: ماتریس lead-field EEG، L، به صورت m×(3n) (مثلاً 33×315) مقداردهی اولیه می‌شود اگر برای اجزای x، y، z هر دوقطبی به طور جداگانه محاسبه شود، یا m×n اگر هر ستون Lij​ نشان‌دهنده پتانسیل از یک دوقطبی j با یک گشتاور از پیش تعریف‌شده (مثلاً شعاعی واحد) باشد. با توجه به "رسم ستون ۷۵ (که نشان می‌دهد چگونه یک دوقطبی خاص بر تمام ۳۳ حسگر تأثیر می‌گذارد)" ، این نشان می‌دهد که L m×n است. توابع Calc\_L و d\_n حیاتی هستند.* 
  + *مسئله احتمالاً فرض می‌کند که هر یک از ۱۰۵ دوقطبی برای این تکلیف برای سادگی در ساخت یک ماتریس m×n برای نمایش یک ستون واحد، دارای یک جهت‌گیری عمدتاً شعاعی هستند. یا، Lij​ را به عنوان پتانسیل از j-امین منبع دوقطبی با فرض یک بردار گشتاور کانونی (مثلاً شعاعی واحد) محاسبه می‌کند.*
* ***تابع*** *d\_n(): این تابع ضرایب dn​ را برای مدل سه‌کره محاسبه می‌کند. این ضرایب به شعاع‌ها (R1​,R2​,R3​) و هدایت‌های (σ1​,σ2​,σ3​) لایه‌های مغز، جمجمه و پوست سر بستگی دارند. فرمول dn​ استاندارد اما طولانی است.*
* ***تابع*** *Calc\_L(): این تابع یک عنصر Lij​ از ماتریس lead-field را محاسبه می‌کند. این تابع مختصات کروی حسگر i (R3​,θi​,ϕi​)، مختصات کروی دوقطبی j (r0j​,θ0j​,ϕ0j​)، بردار گشتاور دوقطبی mj​، و ضرایب dn​ را می‌گیرد. این تابع بسط سری را برای پتانسیل پیاده‌سازی می‌کند.* 
  + *Lij​=4πσ3​1​∑n=1Nmax​​(2n+1)(R3​r0j​​)nR3​dn​(j)​[(m^j​⋅r^0j​)Pn​(cosγij​)+(m^j​⋅t^1,ij​)Pn1​(cosγij​)cosαij​+…]*
  + *که σ3​ هدایت پوست سر است. cosγij​ کسینوس زاویه بین حسگر i و دوقطبی j است. m^j​ بردار گشتاور واحد برای دوقطبی j است. Nmax​ حد برش برای سری است (مثلاً ۵۰-۱۰۰).*
* ***محاسبه اجزای L****: کد برای حسگرها i و دوقطبی‌ها j تکرار می‌شود. برای هر جفت، Lij​ با استفاده از Calc\_L محاسبه می‌شود. یک گشتاور دوقطبی ثابت (مثلاً mj​=[0,0,1] نسبت به جهت شعاعی خود دوقطبی، یا یک جهت‌گیری مختصات جهانی ثابت) باید برای هر دوقطبی j فرض شود تا L یک ماتریس m×n شود که در آن هر ستون مربوط به یک منبع است.*
* ***نمایش****: ستون ۷۵ از L (L[:,74]) رسم می‌شود.*

*توضیح خروجی (Task9.py):*

*اسکریپت یک نمودار (مشابه شکل ۸ در PDF) خروجی می‌دهد که مقادیر ستون ۷۵ ماتریس lead-field EEG، L، را نشان می‌دهد. این نمودار ۳۳ نقطه خواهد داشت که نشان‌دهنده پتانسیل الکتریکی اندازه‌گیری‌شده توسط هر الکترود EEG به دلیل ۷۵امین منبع دوقطبی (با فرض یک گشتاور خاص برای آن) است.*

***پاسخ به سوالات تکلیف ۹****:*

* ***ابعاد L و نمایش****:* 
  + *اگر Lij​ پتانسیل در حسگر i از دوقطبی j باشد (با فرض یک گشتاور کانونی ثابت برای دوقطبی j، مثلاً شعاعی واحد)، آنگاه L m×n است، بنابراین 33×105. این نشان‌دهنده حساسیت هر حسگر به هر منبع دوقطبی (با جهت‌گیری کانونی) است.*
  + *اگر L گشتاورهای دوقطبی دلخواه mj​=[mjx​,mjy​,mjz​] را در نظر بگیرد، آنگاه L m×3n خواهد بود، یعنی 33×315. یک عنصر Li,k​ نشان‌دهنده پتانسیل در حسگر i به دلیل k-امین جزء یک گشتاور دوقطبی (مثلاً mjx​) خواهد بود. پتانسیل Vi​=∑k=13n​Li,k​mk​.*
  + *با توجه به رسم "ستون ۷۵ از L" که به یک دوقطبی اشاره دارد، تفسیر 33×105 برای زمینه این سوال خاص محتمل‌تر است.*
* ***معنای فیزیکی L[i,j]****: L[i,j] نشان‌دهنده پتانسیل الکتریکی اندازه‌گیری‌شده توسط i-امین الکترود EEG است که توسط j-امین منبع دوقطبی تولید می‌شود، با فرض اینکه این دوقطبی دارای قدرت واحد و جهت‌گیری از پیش تعریف‌شده (مثلاً شعاعی) است.*
* ***چگونه L منابع دوقطبی را به اندازه‌گیری‌های حسگر مرتبط می‌کند****: ماتریس lead-field EEG، L، به طور خطی قدرت‌ها/گشتاورهای منبع دوقطبی را به اندازه‌گیری‌های پتانسیل الکتریکی در الکترودهای EEG مرتبط می‌کند. اگر M بردار قدرت‌های منبع دوقطبی (n×1 یا 3n×1) و V بردار پتانسیل‌های الکترود (m×1) باشد، آنگاه V=LM.*
* ***ستون ۷۵ از L نشان می‌دهد****: ستون ۷۵ از L (L[:,74]) توزیع پتانسیل الکتریکی را در تمام ۳۳ الکترود EEG در پاسخ به ۷۵امین منبع دوقطبی (با فرض یک گشتاور واحد خاص برای آن) نشان می‌دهد.*

***تکلیف ۱۰: پتانسیل الکتریکی از یک منبع دوقطبی واحد در EEG***

***هدف****: محاسبه و نمایش پتانسیل الکتریکی در هر الکترود EEG به دلیل یک منبع دوقطبی واحد. رسم سطح سر. (مشابه تکلیف ۵ برای MEG). پارامترهای دوقطبی (موقعیت، گشتاور) باید تعریف شوند (مثلاً مشابه تکلیف ۵: ۷ سانتی‌متر، 45∘,45∘، گشتاور [0,0,1]).*

***توضیح کد (****Task10.py****)****:*

* ***بارگذاری داده‌ها****: مختصات دکارتی حسگر بارگذاری و به کروی (r, theta, phi) تبدیل می‌شوند.*
* ***تعریف پارامترهای دوقطبی****: مختصات کروی یک دوقطبی واحد (مثلاً theta\_0\_dipole, phi\_0\_dipole, radius\_0\_dipole) تعریف و به دکارتی rq\_dipole تبدیل می‌شوند. یک بردار گشتاور دوقطبی q\_dipole (مثلاً [0,0,1]) تعریف می‌شود.*
* ***ثابت‌ها****: پارامترهای مدل سر و هدایت‌ها تنظیم می‌شوند.*
* ***محاسبه پتانسیل‌ها (V)****:* 
  + *پتانسیل الکتریکی Vi​ برای هر یک از m=33 الکترود EEG به دلیل این دوقطبی واحد محاسبه می‌شود.*
  + *این شامل استفاده از توابع Calc\_L و d\_n تعریف‌شده/استفاده‌شده در تکلیف ۹ است. برای هر حسگر i: $V\_i = \text{Calc\_L}(\text{sensor}\_i, \text{dipole}, \text{moment\_vector}, d\_n\text{\_coeffs})$*
  + *نتیجه V یک آرایه از ۳۳ مقدار پتانسیل خواهد بود.*
  + *ساختار Task10.py ممکن است شامل تنظیم n=1 (یک دوقطبی) و محاسبه یک ماتریس lead-field 33×1 L\_single\_dipole باشد. سپس V = L\_single\_dipole[:,0] (اگر گشتاور در L گنجانده شده بود) یا V = L\_single\_dipole @ q\_dipole (اگر L 33×3 برای دوقطبی واحد بود).*
* ***نمایش****:* 
  + *یک نمودار پراکندگی سه‌بعدی از الکترودهای EEG ایجاد می‌شود.*
  + *رنگ هر نقطه الکترود با پتانسیل Vi​ محاسبه‌شده در آن الکترود، با استفاده از یک نقشه رنگی تعیین می‌شود. یک نوار رنگی مقیاس ولتاژ را نشان می‌دهد.*
  + *منبع دوقطبی (مکان و گشتاور) رسم می‌شود (مثلاً با استفاده از ax.quiver).*
  + *سطح سر (کره یا نیمکره) رسم می‌شود.*

*توضیح خروجی (Task10.py):*

*اسکریپت یک نمایش سه‌بعدی (مشابه شکل ۹ در PDF) تولید می‌کند که الکترودهای EEG را روی پوست سر نشان می‌دهد. هر الکترود بر اساس پتانسیل الکتریکی که از دوقطبی واحد مشخص‌شده اندازه‌گیری می‌کند، رنگ‌آمیزی می‌شود. نمودار توزیع فضایی پتانسیل الکتریکی را در سراسر پوست سر که توسط آن دوقطبی تولید می‌شود، نشان می‌دهد. برخلاف میدان MEG، پتانسیل‌های EEG تمایل به پراکندگی و گستردگی بیشتری دارند.*

***تکلیف ۱۱: شبیه‌سازی یک سیگنال EEG متغیر با زمان از یک منبع دوقطبی***

***هدف****: شبیه‌سازی یک سیگنال EEG متغیر با زمان در حسگر شماره ۳۰، معادل تکلیف ۶ برای MEG. از همان دوقطبی (موقعیت، جهت‌گیری) و اندازه متغیر با زمان w(t) استفاده کنید.*

***توضیح کد (****Task11.py****)****:*

* ***تعریف*** *w(t)* ***و بردار زمان****: مشابه تکلیف ۶، تابع w\_func(t) تعریف می‌شود، و یک بردار زمان t و w\_values ایجاد می‌شوند (مثلاً برای ۲ ثانیه با ۱۰۰۰ هرتز).*
* ***بارگذاری lead-field EEG / محاسبه پتانسیل ثابت****:* 
  + *ما به مقدار lead-field L30,dipole​ نیاز داریم که پتانسیل الکتریکی را در حسگر شماره ۳۰ (شاخص ۲۹) به دلیل دوقطبی خاص (در ۷ سانتی‌متر، 45∘,45∘ با جهت‌گیری [0,0,1]) در صورتی که دارای اندازه واحد بود، بدهد.*
  + *این مقدار می‌تواند از ماتریس lead-field کامل L محاسبه‌شده در تکلیف ۹ (اگر برای این پیکربندی دوقطبی خاص به عنوان یکی از ۱۰۵ مورد بود) یا به طور مستقیم برای این دوقطبی واحد و حسگر شماره ۳۰ با استفاده از تابع Calc\_L محاسبه شود. اجازه دهید این Lsensor30,dipole​ باشد.*
  + *بردار جهت‌گیری دوقطبی q تعریف می‌شود (مثلاً [0,0,1]).*
  + *V\_static: پتانسیل ثابت اگر w(t)=1 بود. Vstatic​=Lsensor30,dipole​. (این فرض می‌کند که Lsensor30,dipole​ از قبل جهت گشتاور خاص را در بر می‌گیرد و پتانسیل اسکالر را برای یک گشتاور واحد در آن جهت می‌دهد).*
* ***محاسبه پتانسیل متغیر با زمان****:* 
  + *V\_new = L\_sensor30\_dipole \* w\_values. مقدار lead-field ایستا با w(t) مقیاس‌بندی می‌شود.*
* ***نمایش****:* 
  + *سه زیرنمودار (مشابه شکل ۱۰ در PDF) ایجاد می‌شوند:* 
    1. *w(t) در مقابل زمان.*
    2. *پتانسیل الکتریکی ثابت Vstatic​ در حسگر شماره ۳۰ در مقابل زمان.*
    3. *پتانسیل الکتریکی متغیر با زمان Vnew​ در حسگر شماره ۳۰ در مقابل زمان.*

*توضیح خروجی (Task11.py):*

*اسکریپت یک شکل با سه نمودار تولید می‌کند:*

1. *شکل موج w(t).*
2. *یک خط صاف برای پتانسیل ثابت Vstatic​ در حسگر شماره ۳۰.*
3. *سیگنال EEG شبیه‌سازی‌شده Vnew​(t) در حسگر شماره ۳۰، که یک نسخه مقیاس‌بندی‌شده از w(t) خواهد بود. نمودارها مشابه نمودارهای تکلیف ۶ برای MEG خواهند بود، اما محور y برای نمودارهای دوم و سوم نشان‌دهنده پتانسیل الکتریکی (ولتاژ) به جای میدان مغناطیسی خواهد بود.*

***راهنمای ارسال (شبیه‌سازی‌شده)***

*این گزارش، همراه با اسکریپت‌های پایتون تکمیل‌شده (MEG\_Sens.py, Dipole\_Loc.py, Unit\_Vect.py, task4.py (MEG), task5.py (MEG), task6.py (MEG), Task9.py (EEG), Task10.py (EEG), Task11.py (EEG)، و هرگونه فایل utility\_functions.py)، ارسالی را تشکیل می‌دهد. برای یک آرشیو فشرده از قرارداد نامگذاری فایل FIS\_GeminiAI\_CA1 استفاده می‌شود. تمام کارها با درک تلاش و تحقیق فردی ارائه شده است.*

*این پاسخ به تکلیف کامپیوتری بر اساس PDF ارائه‌شده را تکمیل می‌کند.*