ARRC: Advanced Reasoning Robot Control—Knowledge-Driven Autonomous Manipulation Using Retrieval-Augmented Generation

مقاله ARRC یک سیستم رباتی پیشرفته معرفی می‌کند که هدف آن تبدیل دستورهای زبانی انسانی به کنترل‌های قابل اجرا و ایمن روی بازوی رباتی است. این سیستم تلاش می‌کند فاصله میان توانایی استدلال مدل‌های زبانی و محدودیت‌های فیزیکی و محیطی ربات‌ها را پر کند. ایده اصلی مقاله این است که ربات برای فهم و انجام دستورهای جدید نباید دوباره آموزش داده شود، بلکه باید بتواند از دانش آماده و ساختاریافته استفاده کند. برای رسیدن به این هدف از روش Retrieval-Augmented Generation یا RAG استفاده شده است؛ یعنی سیستم قبل از تولید برنامه حرکتی، دانش مورد نیاز را از یک پایگاه دانش مخصوص ربات بازیابی می‌کند و بعد از آن مدل زبانی با کمک این اطلاعات یک نقشه اجرایی تولید می‌کند. این نقشه به صورت JSON تولید می‌شود تا بتوان آن را کاملاً کنترل، ارزیابی و امن اجرا کرد.

در ابتدا مقاله توضیح می‌دهد که چرا انجام دستورات زبانی توسط ربات به شکل مستقیم کار ساده‌ای نیست. ربات برای گرفتن یک جسم ساده باید چند کار پشت سر هم انجام دهد: پیدا کردن جسم، محاسبه موقعیت سه‌بعدی آن، رسیدن به نقطه‌ای امن در بالای آن، تنظیم زاویه‌ها و سپس گرفتن و جابه‌جایی آن. مدل‌های زبانی بزرگ مانند GPT یا PaLM-E می‌توانند بخشی از این روند را بفهمند اما چون با محدودیت‌های فیزیکی، حد سرعت، شعاع گردش مفاصل و قوانین ایمنی آشنا نیستند، ممکن است برنامه‌های خطرناک یا غیرقابل اجرا بسازند. از سوی دیگر، روش‌های سنتی که فقط بر روی مدل‌سازی و کنترل تمرکز دارند فاقد قدرت استدلال و عمومیت هستند. بنابراین سیستم ARRC ترکیب این دو دنیا است: مدل زبانی برای استدلال و فهم وظیفه، و سیستم رباتی محلی برای کنترل، قوانین ایمنی، محدودیت‌ها و اجرا.

نویسندگان سپس به تفاوت ARRC با پژوهش‌های قبلی می‌پردازند. روش‌هایی مانند RT-1 و RT-2 و PaLM-E بر آموزش گسترده روی داده‌های زیاد تکیه می‌کنند و اگر وظیفه جدید یا محدودیت جدید اضافه شود باید دوباره آموزش ببینند. روش‌های دیگری مانند SayCan سعی می‌کنند از مدل زبانی برای تقسیم وظایف استفاده کنند و سپس از یک مدل دیگر برای بررسی امکان‌پذیری بهره ببرند، اما این روش‌ها همچنان وابسته به مجموعه داده‌های اولیه هستند. همچنین روش‌های retrieval-based که قبلاً وجود داشتند معمولاً فقط قوانین بالادستی را بررسی می‌کردند و به بخش‌های پایین‌دستی مثل بررسی موقعیت واقعی، برخورد با موانع یا خطاهای زمان اجرا اهمیت نمی‌دادند ARRC .در این مقاله سعی دارد همه این بخش‌ها را به صورت یکپارچه در یک سیستم واحد جمع کند تا ربات بتواند هم استدلال زبانی داشته باشد و هم حرکات واقعی را با ایمنی کامل انجام دهد.

مقاله در بخش بعد معماری سیستم را معرفی می‌کند ARRC .از سه بخش بزرگ ساخته شده است: بخش دید، بخش برنامه‌ریزی، و بخش اجرا. در بخش دید از دوربین RealSense D435 به همراه AprilTag استفاده شده تا موقعیت سه‌بعدی اجسام در فضای ربات به صورت کاملاً دقیق مشخص شود. استفاده از AprilTag باعث می‌شود ربات بتواند به شکل قابل اعتماد اجسام را پیدا کند، حتی اگر بخشی از جسم مخفی باشد یا محیط کمی شلوغ باشد. داده‌های به دست آمده از دوربین با استفاده از عمق و هندسه دوربین به مختصات سه‌بعدی تبدیل می‌شود و سپس داخل دستگاه ذخیره می‌شود تا مدل زبانی بتواند از آن استفاده کند.

بخش مهم دوم سیستم، پایگاه دانش است. این پایگاه شامل نمونه‌های کوتاه و ساختاریافته از دانش رباتی است که ربات باید آن‌ها را بداند؛ برای مثال الگوی حرکت برای نزدیک شدن به یک جسم، ارتفاع امن برای قرارگیری بازو روی جسم، فاصله مناسب برای گرفتن، روش‌های عقب‌نشینی امن، دستورهای باز کردن گریپر یا بستن آن، و همچنین قالب‌های وظیفه مانند اسکن، نزدیک شدن، گرفتن و انتقال. در این پایگاه دانش، همچنین قوانین ایمنی، حدود حرکتی و محدودیت‌های سرعت و نیرو ذخیره شده است. هر قطعه دانش به صورت کوتاه نوشته شده تا مدل زبانی بتواند هنگام برنامه‌ریزی به آن مراجعه کند.

هنگامی که کاربر یک دستور زبانی می‌دهد، ابتدا این دستور به یک بردار تبدیل می‌شود و سپس این بردار با داده‌های پایگاه دانش مقایسه می‌شود تا مرتبط‌ترین اسناد پیدا شود. این اسناد که به صورت بردار ذخیره شده‌اند از طریق ChromaDB یا FAISS انتخاب می‌شوند. سپس اسناد انتخاب شده همراه با خلاصه محیط فعلی و قالب JSON برای مدل زبانی ارسال می‌شوند. مدل زبانی با استفاده از این اطلاعات یک برنامه JSON تولید می‌کند که شامل هدف و گام‌های حرکت است. هر گام شامل یک عمل مشخص و پارامترهای محدود شده است؛ برای مثال حرکت به یک نقطه خاص با محدودیت سرعت یا حرکت به بالای جسم با حداکثر ارتفاع.

در مرحله بعد سیستم باید این برنامه را بررسی کند. این بررسی برای جلوگیری از اشتباهات مدل زبانی است. در این بخش موقعیت‌های انتخاب‌شده با اطلاعات لحظه‌ای دوربین بررسی می‌شود تا اگر جسم جابه‌جا شده باشد برنامه اصلاح شود. همچنین اگر یک گام غیرایمن یا غیرقابل انجام باشد، سیستم آن را رد می‌کند. فقط گام‌های قابل اجرا اجازه ورود به مرحله اجرا را پیدا می‌کنند.

بخش اجرا، گام‌ها را به دستورات سطح پایین تبدیل می‌کند و آن‌ها را از طریق SDK بازوی xArm به اجرا می‌گذارد. این مرحله با قوانین ایمنی سخت‌گیرانه همراه است. این قوانین از جمله محدودیت سرعت، محدودیت نیرو و گشتاور گریپر، جلوگیری از عبور از محدوده کاری، جلوگیری از برخورد با سطح میز، محدودیت برای زمان انجام هر گام و حتی محدودیت تعداد تلاش‌ها برای گرفتن اجسام است. اگر ربات احساس کند جسم گیر کرده یا بار اضافی وارد شده است گریپر به صورت خودکار باز می‌شود. این رفتار از خرابی سخت‌افزار جلوگیری می‌کند.

مقاله در بخش بعد فرآیندهای الگوریتمی سیستم را توضیح می‌دهد. الگوریتم اصلی برنامه‌ریزی RAG توضیح می‌دهد که دستور کاربر چگونه به بردار تبدیل می‌شود و چگونه با پایگاه دانش مقایسه شده و اسناد مناسب انتخاب می‌شوند. سپس اسناد همراه با وضعیت محیط به مدل زبانی ارسال می‌شود تا برنامه JSON تولید شود. علاوه بر آن الگوریتم اسکن دو مرحله‌ای ربات توضیح داده شده است. در این الگوریتم ابتدا ربات از مسیرهای افقی برای پیدا کردن اجسام استفاده می‌کند. اگر جسم پیدا نشود ربات از یک اسکن قوسی کمک می‌گیرد. این روش باعث می‌شود ربات بتواند اجسام پنهان یا اجسامی که در زاویه دید محدود قرار دارند پیدا کند.

در بخش آزمایش‌ها مقاله یک بازوی رباتی xArm 850 مجهز به گریپر Dynamixel را استفاده کرده است. دوربین RealSense برای تشخیص اجسام و یک سیستم کامپیوتری برای پردازش داده‌ها در نظر گرفته شده است. محیط آزمایش شامل میز و چند جسم روزمره مانند بطری، جعبه، ابزارها و پیچ‌گوشتی بوده است. این اجسام در موقعیت‌های تصادفی اما کنترل شده قرار گرفته‌اند تا عملکرد سیستم ارزیابی شود.

آزمایش‌ها شامل سه وظیفه بوده است: اسکن، نزدیک شدن به جسم و انجام کامل عملیات برداشتن و جابه‌جایی. در آزمایش‌ها سیستم توانسته در تمام تلاش‌ها حداقل یک جسم را پیدا کند. درصد موفقیت نزدیک شدن به جسم نسبتاً بالا و حدود ۸۷ درصد بوده است. همچنین در هشت تلاش از ده تلاش سیستم توانسته عمل برداشتن و جابه‌جایی جسم را با موفقیت انجام دهد. زمان اجرای برنامه‌ها نیز در حدود یک ثانیه بوده است که نشان می‌دهد سیستم علاوه بر دقت از سرعت مناسب برخوردار است.

یکی از آزمایش‌ها نشان داده است که ربات می‌تواند در صورت مخفی بودن جسم، استراتژی اسکن خود را تغییر دهد. برای مثال در مورد پیچ‌گوشتی که با زاویه قرار گرفته بود ربات ابتدا آن را ندید، سپس به صورت خودکار اسکن قوسی انجام داد و پس از دیدن جسم توانست آن را بردارد. این قابلیت نشان می‌دهد مدل زبانی و سیستم RAG می‌توانند در برابر شرایط جدید و پیش‌بینی نشده استدلال کنند.

در نهایت مقاله اشاره می‌کند که ARRC از نظر ایمنی و انعطاف‌پذیری نسبت به روش‌های قبلی بهتر است. اما همچنان کمبودهایی مثل نبود حسگرهای لمسی، ناتوانی در انجام وظایف پیچیده مانند پیچ‌کردن و محدودیت در پایگاه دانش وجود دارد. برای آینده قرار است حسگرهای بیشتر، محیط‌های پیچیده‌تر و مدل‌های محلی با سرعت بیشتر به سیستم اضافه شود.