Abstract

La robotica è una delle tecnologie più importanti della società moderna. I progressi nei campi dell'informazione, dell'elettronica e della meccanica ci permettono di costruire e programmare macchine per svolgere compiti nei contesti più disparati, come l'industria, la chirurgia e il settore aerospaziale.
In particolare, nella produzione manifatturiera, i robot vengono utilizzati principalmente per eseguire lavori ripetitivi e potenzialmente dannosi per l'uomo, come l'assemblaggio, la saldatura e la movimentazione di materiali pesanti o pericoli. Questo è possibile grazie alla ben nota robustezza meccanica e alla capacità di ripetere gli stessi movimenti con alta precisione e accuratezza.

In passato, i sistemi robotici venivano utilizzati in contesti con forti limitazioni date dal fatto che l'ambiente era chiuso e noto a priori. Negli ultimi decenni, grazie anche allo sviluppo del paradigma dell'Industria 4.0, i robot sono immersi in contesti più flessibili, rappresentati da ambienti non noti o parzialmente noti a priori, dove devono **coesistere** e **cooperare** con gli esseri umani, risolvendo diversi compiti **dinamici** [1] (ad esempio, prelevare un oggetto richiesto la cui posizione non è nota a priori).

In questo scenario, le caratteristiche desiderate di tali sistemi robotici sono:

(a) Adattabilità a nuove condizioni, ossia il sistema deve essere in grado di adattarsi facilmente ai cambiamenti delle condizioni di contesto, mostrando comportamenti *"intelligenti"* per affrontare questi nuovi scenari andando a risolvere il problema di interesse.

(b) Adattabilità a nuovi compiti, ossia il sistema deve essere in grado di adattarsi facilmente sia a nuove varianti di un compito conosciuto che a compiti completamente nuovi, sfruttando l'esperienza per inferire le azioni necessarie a risolverli.

Questi requisiti possono risultare difficili da raggiungere con le tecniche tradizionali di programmazione robotica basate su politiche di controllo definite manualmente. Queste tecniche convenzionali richiedono spesso un'analisi meticolosa delle dinamiche del processo, la costruzione di un modello analitico e la derivazione di una legge di controllo che soddisfi criteri di progettazione specifici. Questo processo di progettazione può risultare tedioso richiedendo tempo e soprattutto particolarmente costoso, soprattutto quando vengono utilizzati sistemi di percezione avanzati (telecamere, microfoni, sensori di movimento) per dedurre lo stato dell'ambiente (come la posizione di un oggetto desiderato) e le intenzioni dell'operatore umano.

Al contrario, sono stati compiuti progressi significativi sfruttando le *tecniche di apprendimento*, in cui la politica di controllo viene dedotta dai dati. Questi dati possono essere generati sia tramite **l'esperienza dell'agente** [2] che tramite **dimostrazioni di esperti** [3].
Nel caso dell'esperienza dell'agente, c'è una procedura *trianl-and-error* in cui la politica di controllo genera azioni eseguite da un agente, che interagisce con l'ambiente. I parametri vengono poi adattati in base all'efficacia delle azioni, tenendo conto del loro impatto sull'ambiente rispetto al compito da risolvere. Nel caso delle dimostrazioni di esperti, i parametri della politica di controllo vengono direttamente adattati utilizzando un dataset contenente esempi dell'esecuzione del compito. L'obiettivo qui è replicare i compiti osservati nel dataset.

Dato questo contesto, la tesi si inserisce nel contesto del *Learning from Demonstration* (LfD), un approccio di apprendimento basato su dimostrazioni di esperti. Rispetto i requisiti di adattabilità, la tesi si concentra su un aspetto specifico dell'LfD, denominato *Multi-Task LfD*. In questo caso, la politica di controllo non viene addestrata per eseguire un singolo compito (ad esempio, prendere un oggetto) con l'obiettivo di generalizzare rispetto a diversi oggetti e condizioni iniziali [4, 5].

Piuttosto, viene addestrata per gestire varie varianti di uno specifico compito (ad esempio, prendere un oggetto da diverse posizioni possibili) [6] o persino compiti completamente diversi (ad esempio, una singola politica di controllo che risolve sia compiti di pick-and-place sia compiti di assemblaggio) [7, 8]. In questo caso, l'obiettivo è generalizzare non solo rispetto agli oggetti manipolati e alle condizioni iniziali, ma anche rispetto ai compiti stessi. Questo significa che, sfruttando l'ipotesi del *knowledge-sharing*, possiamo ottenere un sistema in grado di risolvere nuove variazioni.

In questo scenario, la procedura di apprendimento è molto più complessa poiché è necessario includere e definire il **segnale di condizionamento** (ossia, il segnale che informa la politica sul compito da eseguire, l'oggetto da manipolare e la posizione di destinazione). Inoltre, l'ambiente può contenere diversi **oggetti distrattori** (ad esempio, oggetti che potrebbero essere manipolati ma che non sono di interesse per una determinata variazione del compito).

Per quanto riguarda il segnale di condizionamento, si possono definire almeno due approcci. Il primo rappresentato da una descrizione in linguaggio naturale del compito da eseguire [9, 10, 7], il secondo è rappresentato da una dimostrazione video [6, 8].
Nel primo caso, il compito è descritto utilizzando frasi che specificano i dettagli del compito, come "Take the red bot and place into the first bin". Fornita la frase in ingresso, il sistema deve essere in grado di dedurre l'intento del compito (ossia, l'operazione di presa e posizionamento) e l'oggetto di interesse (ad esempio, la scatola rossa da prendere e il primo contenitore per il posizionamento) e correlare queste informazioni con l'ambiente e lo stato del robot per controllarlo in modo efficace.
Nel secondo caso, un altro agente (sia esso un robot o un operatore umano) esegue il compito in una diversa configurazione ambientale, registra questa esecuzione e fornisce il video come input alla politica di controllo. La politica di controllo deve quindi dedurre l'intento dal video (ossia, il compito da eseguire, l'oggetto da manipolare e lo stato finale) e controllare il robot per completare il compito in base allo stato dell'agente, allo stato dell'ambiente e al compito comandato.

Prendendo inspirazione da come gli esseri umani possono imparare a replicare compiti semplicemente osservandone l'esecuzione di questi ultimi, l'obiettivo principale di questa tesi è sviluppare un sistema in grado di replicare i compiti mostrati in una dimostrazione video. Questo comporta affrontare sfide legate all'estrazione di informazioni rilevanti per il compito dal video, come l'identificazione dell'oggetto manipolato e la sua posizione finale.

Per quanto riguarda il problema legato alla presenza di oggetti distrattori, in generale questi sono oggetti che non vengono mai considerati in operazioni di manipolazione, semplificando di molto il problema. Tuttavia, nel contesto proposto in questa tesi, il problema è ulteriormente enfatizzato dal fatto che il significato semantico di oggetto di interesse o distrattore è definito a run-time dal comando stesso. Questo significa che se la configurazione iniziale è composta da quattro oggetti (ad esempio, quattro box di colore diverso), sulla base del comando dato al robot un determinato oggetto può diventare o meno di interesse.

Il principale contributo di questa tesi è quello di svillupare un sistema che sia robusto alla presenza di distrattori all'interno della scena. Nello specifico, un problema chiave identificato nella letteratura esistente è la **target missidentification**, questo significa che la politica di controllo appresa genera traiettorie valide, permettendo al robot di raggiungere, prendere e posizionare oggetti, ma manipolando l'oggetto sbagliato.

Per risolvere questo problema, sono state fatte due considerazioni principali:

(1) Le architetture proposte in letteratura sono prevalentemente **end-to-end**, questo significa che traducono input ad alta dimensionalità (immagini) nelle corrispondenti azioni (posa del gripper rispetto ad un frame di riferimento). Con questo approccio, il modello deve imparare una rappresentazione implicita che codifica sia l'obiettivo del compito che lo stato corrente dell'ambiente, compresa la posizione dell'oggetto target.

(2) La procedura di apprendimento ottimizza una funzione di errore che si focalizza esclusivamente sull'azione, que-

sto significa che il sistema durante l'addestramento ha come obiettivo quello di generare in media le stesse azioni presenti nel dataset. Questa procedura può portare il sistema al non focalizzarsi sulla codifica di informazioni di interesse come la posizione degli oggetti.

Questi due fattori possono portare a una politica di controllo che non riesce a guidare efficacemente il robot verso l'oggetto target. In particolare, è stato osservato che le fasi iniziali dell'esecuzione della traiettoria sono cruciali. Infatti, anche piccoli errori durante questi primi passi possono portare il robot al raggiungimento e la presa dell'oggetto sbagliato.

Basandosi su queste considerazioni, questa tesi esplora lo sviluppo di un'architettura **modulare**, in contrapposizione agli approcci end-to-end proposti in letteratura. Questa architettura prevede moduli specificamente progettati per ragionare sulle zone di interesse (ad esempio, la posizione dell'oggetto target e la posizione finale di posizionamento). Quindi, una volta individuate queste zone di interesse, attraverso la generazione di bounding-box, questi possono essere integrati nell'input del modulo di controllo, che ora riceve anche informazioni a bassa dimensionalità, come la posizione dell'oggetto target, che possono essere più facilmente utilizzate durante il processo di apprendimento, soprattutto alla luce della perdita centrata sull'imitazione dell'azione.

Per eseguire questo ragionamento esplicito, è stato sviluppato un *Conditioned Object Detector* (COD). Questo modulo, dato in input il video della dimostrazione e l'osservazione corrente dell'agente, predice il bounding-box relativo all'oggetto target e alla posizione finale.

La procedura di apprendimento viene quindi suddivisa in due fasi. La prima fase prevede l'addestramento del modulo COD, focalizzandosi sulla risoluzione dei problemi cognitivi di detection. La seconda fase prevede l'addestramento della *Object-Conditioned Control Policy* (OCCP), che si concentra sulla risoluzione del problema di controllo sfruttando le informazioni posizionali generati dal COD.

Il sistema finale è stato ampiamente testato in simulato, dove è possibile generare scenari e collezionare traiettorie per

il dataset. La validazione del sistema si è conclusa attraverso il testing dei metodi proposti su una piattaforma robotica reale.

Per quanto riguarda l'ambiente simulato, il sistema è stato valutato sia in scenari definiti **multi-variation single-task** che in scenari definiti **multi-variation multi-task**, considerando quattro compiti: Pick-Place, Nut-Assembly, Stack-Block e Press-Button. Ogni task è caratterizzato dalla presenza di diverse varianti definite sulla base dell'oggetto manipolato e del suo stato finale. Per esempio, nel task di Pick-Place sono presenti 4 box e 4 bin, le variazioni sono rappresentate dalle possibili combinazioni di box da prelevare e bin dove eseguire il placing.

I task selezionati sono caratterizzate dalla presenza sia di caratteristiche comuni, ma anche di caratteristiche specifiche. Ad esempio, il compito di Nut-Assembly comporta una manipolazione precisa, mentre il Pick-Place può essere risolto in modo molto più approssimativo.

Nel complesso, i metodi proposti hanno dimostrato comportamenti molto promettenti e un miglioramento generale rispetto ai metodi di base che non includono il ragionamento sugli oggetti. Ciò dimostra che risolvere compiti di manipolazione con un approccio orientato agli oggetti può essere un paradigma efficace per i problemi di LfD. Inoltre, questo approccio fornisce informazioni interpretabili all'utente finale, poiché i riquadri di delimitazione previsti possono essere interpretati come le posizioni verso cui si muoverà il robot.

In conclusione, il metodo proposto è stato testato anche in un ambiente reale, dove la complessità del problema è aumentata dalla presenza di dati scarsi e rumorosi raccolti tramite teleoperazione. Anche in queste condizioni, il metodo proposto ha dimostrato la sua efficacia nell'affrontare sia i problemi cognitivi che quelli di controllo. Ciò conferma che l'informazione legata all'oggetto di interesse può essere applicata con successo in scenari reali, consentendo lo sviluppo di un sistema affidabile nonostante dati limitati e rumorosi.

Bibliografia

[1] S. Bini, G. Percannella, A. Saggese, and M. Vento, "A multitask network for speaker and command recognition in industrial environments," *Pattern Recognition Letters*, vol. 176, pp. 62–68, 2023.

[2] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[3] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, J. Peters *et al.*, "An algorithmic perspective on imitation learning," *Foundations and Trends® in Robotics*, vol. 7, no. 1-2, pp. 1–179, 2018.

[4] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel, "Deep imitation learning for complex manipulation tasks from virtual reality teleoperation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 5628–5635.

[5] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, "What matters in learning from offline human demonstrations for robot manipulation," in *Conference on Robot Learning*. PMLR, 2022, pp. 1678–1690.

[6] S. Dasari and A. Gupta, "Transformers for one-shot visual imitation," in *Conference on Robot Learning*. PMLR, 2021, pp. 2071–2084.

[7] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath,

I. Mordatch, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. S. Ryoo, G. Salazar, P. R. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. T. Tran, V. Vanhoucke, S. Vega, Q. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich, "RT-1: robotics transformer for real-world control at scale," in *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, K. E. Bekris, K. Hauser, S. L. Herbert, and J. Yu, Eds., 2023. [Online]. Available: https://doi.org/10.15607/RSS.2023.XIX.025

[8] Z. Mandi, F. Liu, K. Lee, and P. Abbeel, "Towards more generalizable one-shot visual imitation learning," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2434–2444.

[9] S. Stepputtis, J. Campbell, M. Phielipp, S. Lee, C. Baral, and H. Ben Amor, "Language-conditioned imitation learning for robot manipulation tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 13 139–13 150, 2020.

[10] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard, "Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks," *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, no. 3, pp. 7327–7334, 2022.