Online appendix for the paper

Manipulation of Articulated Objects using Dual-arm Robots via Answer Set Programming

RICCARDO BERTOLUCCI
University of Calabria, Italy
(*e-mail:* bertolucci@mat.unical.it)

ALESSIO CAPITANELLI
Teseo srl, Italy
(*e-mail:* alessio.capitanelli@teseotech.com)

CARMINE DODARO, NICOLA LEONE
University of Calabria, Italy
(*e-mail:* {dodaro,leone}@mat.unical.it)

MARCO MARATEA, FULVIO MASTROGIOVANNI
University of Genoa, Italy
(*e-mail:* {marco.maratea,fulvio.mastrogiovanni}@unige.it)

MAURO VALLATI
University of Huddersfield, UK
(*e-mail:* m.vallati@hud.ac.uk)

submitted 1 January 2003; revised 1 January 2003; accepted 1 January 2003

Appendix A

Figures 1 and 2 report the MAES encoding. Rules m_1, m_2 and m_3 are related to the selection of possible macro actions in this model: m_1 is the macro that locates the joint that has to be moved to the centre of the manipulation space and then grasp two links, m_2 selects at which orientation pre-grasped links have to be moved and then release them. Rule m_3 selects two links to grasp, the orientation at which one of the link must be moved and, finally, it states that those two link must be released.

Moreover, r_{m1} to r_{m5} are used to state, respectively, which joint will be in the centre of the workspace at `T+1` time step, which links are in the robots hand and which gripper grasped which link. Rules r_{m6} to r_{m10} are used to ensure that the selected angle is one of the possible angle and that the angle is always between 0 and 360 degrees. Instead, rules r_{m10} and r_{m11} are used to compute which links are affected by the movement of the link selected by m_2, whereas, rules r_{m13} update the orientation of the selected link, and rules r_{m14} and r_{m15} state that the grasped links are free in the next time step if m_2 is selected. Rules from r_{m16} to r_{m25} have the same purpose as rules from r_{m6} to r_{m15} but with respect to m_3. Then, rules r_{m26} to r_{m29} are used to ensure that only one macro is selected at each time step. Rules r_{m30} to r_{m33} are used to control the propagation overtime of the atoms `free`, `grasped`, `in_hand` and `hasAngle` with respect to the selected action. Finally, rule r_{m35} states the the goal must be reached.

Consider Example **??**, in which we wrote down $pre(r_{16,18})$, $del(r_{16,18})$ and $add(r_{16,18})$, where

```
m1    {linkToCentral_take(L1,L2,J1,G1,G2,T)} :-link(L1),link(L2),joint(J1),
          gripper(G1),gripper(G2),time(T),free(G1,T),free(G2,T),
          not in_centre(J1,T),not in_hand(L1,T),not in_hand(L2,T),
          L1<>L2,G1<>G2,connected(J1,L1),connected(J1,L2)
rm1   in_centre(J1,T+1):-linkToCentral_Grasp(_,_,_,J1,_,T),T<timemax+1.
rm2   in_hand(L,T+1):-linkToCentral_Grasp(L,_,_,_,_,T),T<timemax+1.
rm3   in_hand(L,T+1):-linkToCentral_Grasp(_,L,_,_,_,T),T<timemax+1.
rm4   grasped(G,L,T+1):-linkToCentral_Grasp(L,_,_,G,_,T),T<timemax+1.
rm5   grasped(G,L,T+1):-linkToCentral_Grasp(_,L,_,_,G,T),T<timemax+1.

m2    {changeAngle_release(L1,L2,J1,G1,G2,A1,A2,T)1}:-link(L1),link(L2),
          joint(J1),gripper(G1),gripper(G2),angles(A1),angles(A2),
          in_centre(J1,T),not free(G1,T),not free(G2,T),L1<>L2,G1<>G2,
          connected(J1,L1),connected(J1,L2),grasped(G1,L1,T),
          in_hand(L1,T),in_hand(L2,T),hasAngle(L1,A2,T),
          grasped(G2,L2,T).
rm6   ok(L1,L2,A,Ai,T):-changeAngle_release(L1,L2,_,_,_,A,Ai,T),
          F1=(A+granularity)\360,F2=(Ai\360),F1 = F2,A < Ai.
rm7   ok(L1,L2,A,Ai,T):-changeAngle_release(L1,L2,_,_,_,A,Ai,T),
          F1=(Ai+granularity)\360,F2=(A\360),F1 = F2,A > Ai.
rm8   ok(L1,L2,A,0,T) :-changeAngle_release(L1,L2,_,_,_,A,Ai,T),
          A = 360-granularity.
rm9   ok(L1,L2,0,A,T) :-changeAngle_release(L1,L2,_,_,_,A,Ai,T),
          A = 360-granularity.
rm10  :- changeAngle_release(L1,L2,_,_,_,A,Ai,T),not ok(L1,L2,A,Ai,T).
rm11  affected(L,An,Ac,T):-changeAngle_release(L1,L2,_,_,_,A,Ap,T),
          angles(An),hasAngle(L,Ac,T),An = |(Ac + (A-Ap)) + 360|\360,
          L>L1,L1>L2.
rm12  affected(L,An,Ac,T):-changeAngle_release(L1,L2,_,_,_,A,Ap,T),
          angles(An),hasAngle(L,Ac,T),An = |(Ac + (A-Ap)) + 360|\360,
          L<L1,L1<L2.
rm13  hasAngle(L1,A1,T+1):-changeAngle_release(L1,L2,_,_,_,A1,_,T),
           T<timemax+1.
rm14  free(G,T+1):-changeAngle_release(_,_,_,G,_,_,_,T),T<timemax+1.
rm15  free(G,T+1):-changeAngle_release(_,_,_,_,G,_,_,T),T<timemax+1.
```

Figure 1: Macro Action Extended Scenario (MAES) encoding (Part 1).

$r_{16,18}$ corresponds to m_1. Here it is possible to outline more precisely the meaning of $del(r_{16,18})$ and $add(r_{16,18})$.
Therefore, since

$$\begin{aligned} del(r_{16,18}) = &\quad \{\texttt{free(G,T)}\} \\ add(r_{16,18}) = &\quad \{\texttt{in\_centre(J1,T)}, \texttt{in\_hand(L1,T)}, \texttt{in\_hand(L2,T)}, \\ &\quad \texttt{grasped(G1,L1,T)}, \texttt{grasped(G2,L2,T)}\}. \end{aligned}$$

it is possible to notice that $del(r_{16,18})$ brings to the definition of r_{m30} in which we state that the atom `free(G,T+1)` must not be propagated if the macro m_1 is selected. Moreover, $add(r_{16,18})$ brings to the definition of the rules from r_{m1} to r_{m5} in which we state that at the next step the atoms `in_centre(J1,T+1)`, `in_hand(L,T+1)`, `grasped(G,L,T+1)` must be true if the macro m_1 is selected.

```
m3    {grasp_changeAngle_release(L1,L2,J1,A1,A2,G1,G2,T)}:- link(L1),
          link(L2),joint(J1),angles(A1),angles(A2),gripper(G1),
          gripper(G2),time(T),in_centre(J1,T),free(G1,T),free(G2,T),
          connected(J1,L1),connected(J1,L2),in_centre(J1,T),
          hasAngle(L1,A2,T),time(T),L1<>L2,G1<>G2.
rm16  ok(L1,L2,A,Ai,T):-grasp_changeAngle_release(L1,L2,_,A,Ai,_,_,T),
          F1=(A+granularity)\360,F2=(Ai\360),F1 = F2,A < Ai.
rm17  ok(L1,L2,A,Ai,T):-grasp_changeAngle_release(L1,L2,_,A,Ai,_,_,T),
          F1=(Ai+granularity)\360,F2=(A\360),F1 = F2,A > Ai.
rm18  ok(L1,L2,A,0,T) :-grasp_changeAngle_release(L1,L2,_,A,Ai,_,_,T),
          A = 360-granularity.
rm19  ok(L1,L2,0,A,T) :-grasp_changeAngle_release(L1,L2,_,A,Ai,_,_,T),
          A = 360-granularity.
rm20  :- grasp_changeAngle_release(L1,L2,_,A,Ai,_,_,T),
          not ok(L1,L2,A,Ai,T).
rm21  affected(L,An,Ac,T):-grasp_changeAngle_release(L1,L2,_,A,Ap,_,_,T),
          angles(An),hasAngle(L,Ac,T),An = |(Ac + (A-Ap)) + 360|\360,
          L>L1,L1>L2.
rm22  affected(L,An,Ac,T):-grasp_changeAngle_release(L1,L2,_,A,Ap,_,_,T),
          angles(An),hasAngle(L,Ac,T),An = |(Ac + (A-Ap)) + 360|\360,
          L<L1,L1<L2.
rm23  hasAngle(L1,A1,T+1):-grasp_changeAngle_release(L1,L2,_,A1,_,_,_,T),
          T<timemax+1.
rm24  free(G,T+1):-grasp_changeAngle_release(_,_,_,_,_,G,_,T),T<timemax+1.
rm25  free(G,T+1):-grasp_changeAngle_release(_,_,_,_,_,_,G,T),T<timemax+1.

rm26  action(T, linkToCentral_Grasp(L1,L2,J1,G1,G2,T)) :-
          linkToCentral_Grasp(L1,L2,J1,G1,G2,T).
rm27  action(T, changeAngle_release(L1,L2,J1,G1,G2,A1,A2,T)) :-
          changeAngle_release(L1,L2,J1,G1,G2,A1,A2,T).
rm28  action(T, grasp_changeAngle_release(L1,L2,J1,A1,A2,G1,G2,T))
          :- grasp_changeAngle_release(L1,L2,J1,A1,A2,G1,G2,T).
rm29  :- time(T), #count{Z : action(T,Z)} != 1.
rm30  free(G,T+1) :- free(G,T), not linkToCentral_Grasp(_,_,_,G,_,T),
           not linkToCentral_Grasp(_,_,_,_,G,T), T < timemax+1.
rm31  in_hand(L,T+1) :- in_hand(L,T),
          not changeAngle_release(L,_,_,_,_,_,_,T),
          not changeAngle_release(_,L,_,_,_,_,_,T),
          not grasp_changeAngle_release(L,_,_,_,_,_,_,T),
          not grasp_changeAngle_release(_,L,_,_,_,_,_,T),
          T < timemax+1.
rm32  grasped(G,L,T+1) :- grasped(G,L,T), link(L),
          not changeAngle_release(L,_,_,G,_,_,_,T),
          not changeAngle_release(_,L,_,_,G,_,_,T),
          not grasp_changeAngle_release(L,_,_,_,_,G,_,T),
          not grasp_changeAngle_release(_,L,_,_,_,_,G,T),
          T < timemax+1.
rm33  hasAngle(L,A,T+1) :- affected(L,A,_,T), T < timemax+1.
rm34  hasAngle(L,A,T+1) :- hasAngle(L,A,T),
          not changeAngle_release(L,_,_,_,_,_,_,T),
          not grasp_changeAngle_release(L,_,_,_,_,_,_,T),
          not affected(L,_,_,T), T < timemax+1.

rm35  :- goal(J,A), not hasAngle(J,A,timemax).
```

Figure 2: Macro Action Extended Scenario (MAES) encoding (Part 2).