In: IEEE/RSJ 2012 International Conference on Intelligent Robots and Systems, Oct. 7–12, 2012, Vilamoura, Algarve, Portugal

Proactive premature-intention estimation for intuitive human robot collaboration

Muhammad AWAIS and Dominik HENRICH
Lehrstuhl für Angewandte Informatik III (Robotik und Eingebettete Systeme)
Universität Bayreuth, D-95440 Bayreuth, Germany
{Muhammad.Awais | Dominik.Henrich}@uni-bayreuth.de

***Abstract*— For effective collaboration between two humans, they are required to adapt to each other according to their apparent behavior in the real time. Similarly for intuitive human-robot collaboration the robot is required to adapt to the human intention, apparent from his behavior. In this paper, we introduce a probability based approach that helps the robot to adapt to the human behavior and to react proactively in the intermixed human intentions scenarios. The robot can either wait for disambiguation of the intention, requiring extra human actions or he can proactively react depending on his previous knowledge about the human behavior. The adaptive proactivity is achieved with the help of local transition weights for transitions between the neighbouring states in a finite state machine. These finite state machines model the human intentions. We also introduce the update of the intention recognition triggers for the intentions. The online intention-trigger update corresponds to the online selection of the specific state of a finite state machine as the end state of the finite state machine. The online update of the intention-triggers makes the robot more proactive in the direction of premature intention estimation. Thus the local transition weights and premature intention estimation makes the human robot collaboration more intuitive.**

***Index Terms*— Premature-intention, Local transition weight, Trigger state.**

I. INTRODUCTION

Proactivity is an important aspect for effective cooperation. The humans working on a common task are required to intuitively collaborate with each other. They are required to be proactive towards each other for intuitive collaboration. It is equally important in the human-robot interaction that the robot should be proactive according to the collaborating human, depending on the current situation. For being proactive, the robot needs to recognize the intention of cooperating human as early as possible. The robot is also required to adapt to the human intentions to be proactive.

In literature there exist different approaches concerning the proactive behavior of the robot. In [7] proactive planning for a mobile robot is proposed. The proactive planning concerns the planning of future actions based on the action that is currently being performed. The proactive planning of actions can only be performed in the proposed approach if the result of currently performed action is predictable. The approach proposed in [1] corresponds to agent planning in a dynamic environment. The agent is created to act in the virtual world designed for a computer game. They described a layered approach for agent actions in the dynamic environment. A prioritization approach is presented in [5]. The approach proposes a hierarchical agent model. The hierarchy consists of four layers, i.e., goal selector, planner, scheduler, and executor. The change in the priorities of the goals is performed to increase the satisfaction level. The goal prioritization and reprioritization relates to learning of the priorities of goal. An approach in [8] described how the robot can plan proactively to make its navigation safe. The approach presents a computing method for maximum velocity of the robot over a planned trajectory. The proactivity corresponds to speed adaption with respect to any number of mobile objects with known maximum possible velocity. The simulating robot described in [6] is an interactive interface. The idea corresponds to the two humans communicating over a telephone line. Instead of using the telephone, the robots are used to communicate the information between the two persons. The approach in [4] proposed the proactive behavior of the robot by activity monitoring and the constraints. The described behavior warns of an operation, forgotten by the humans. A behavioral generation mechanism is proposed in [10]. The proactive behavior is produced by two layers which are behavior planning and behavior generation. The behavior corresponds to different posses and movements. The term proactive is very much primitive as compared to the proactiveness of a human. The discussed approaches do not directly correspond to strict human-robot interaction. The proactive action selection proposed in [9] describes the proactive robot response concerning human-robot interaction. The robot selects actions concerning the known human intention. The proactive action selection is performed given the estimate of the human intention. However, there are vital differences between [9] and presented approach. The approach presented in [9] assumes that the intention estimates are given, uses a number of intentions for proactive reaction, focuses on the actions rather on the proactive action selection and considers all the intentions without considering their relevance with respect to the current situation. The approach in [9] also does not provide a confusion resolution if the intention estimates consists of conflicting intentions.

The presented approach recognizes the human intention as early as possible for being proactive and resolves the conflicting intentions by adapting proactively to the most likely premature intention.

The rest of the paper is organized as follows. The Section II describes briefly the intention recognition using probabilistic state machine [3]. The Section III describes the online trigger (end state) determination of a finite state machine. The local transition weights are discussed in Section IV. The Section V describes the experiments performed and the discussion of the results. We conclude in the Section VI with the comments on the presented approach.

II. Intention Recognition Using State Machines

Each unique human intention is represented by a distinct Finite State Machine (FSM). A FSM models the action sequence corresponding to a unique human intention. Each FSM carries a probabilistic weight. The weight represents how closely the intention represented by the FSM relates to the currently estimated human intention. If the weight is high then the FSM closely relates to the currently estimated human intention and vice versa. A general FSM is shown below in the Fig. 1. Each action a_{ji} where $j = 1,...,m$ and $i = 1,...,n$ has a probability value $P(a_j \mid S_i)$ at a state S_i. The probability value $P(a_j \mid S_i)$ describes how likely an action a_j is for the state S_i of a FSM. The action a_{ki} represents an action that has highest probability $P(a_k \mid S_i)$ for the state S_i and the state transition only occurs if a_{ki} occurs as shown in Fig. 1.

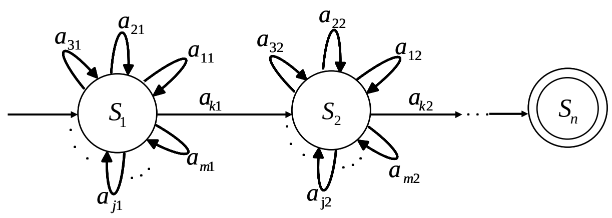

Fig 1. Flow of intention recognition algorithm

The general description of a FSM is given below in the Fig. 2. Each $FSM = \langle Q, \Sigma, , q_0, F, \delta \rangle$ is a tupple that contains a set $Q = \{S_1, S_2, S_3, ..., S_n\}$ that represent the number of states in a FSM. The set $\Sigma = \{a_1, a_2, a_3, ..., a_m\}$ represents the possible actions for a state $S_i \in Q$.

$$FSM = < Q, \Sigma, q_o, F, \delta >$$
$$Q = \{S_1, S_2, S_3, ..., S_n\}$$
$$\Sigma = \{a_1, a_2, a_3, ..., a_m\}$$
$$\forall S_i \in Q \wedge \forall a_x \in \Sigma \text{ it holds that } \sum_{x=1}^{m} p(a_x \mid S_i) = 1$$
$$\forall S_i : \exists a_k \in \Sigma : \underset{j=1, j \neq k}{\overset{m}{\forall}} a_j \text{ it holds that } \left[p(a_k \mid S_i) > p(a_j \mid S_i) \right]$$
$$\delta : Q \text{ x } \Sigma \rightarrow Q$$
$$\delta(S_i, a_j) = S_i \text{ and } \delta(S_i, a_k) = S_{i+1} \quad i = 1,...,n$$
$$q_0 = S_1$$
$$F = \{S_n\}$$

Fig 2. General description of a Finite State Machine

The sum of probabilities of all the actions $\sum_{x=1}^{m} P(a_x \mid S_i)$ for a state $S_i \in Q$ adds up to 1, i.e.

$$\forall S_i \in Q \ \wedge \forall a_x \in \Sigma \ \text{ it holds that } \sum_{x=1}^{m} P(a_x \mid S_i) = 1$$

For each state $S_i \in Q$ there exists an action a_k such that the probability of the action $P(a_k \mid S_i)$ is greater than all the other actions $\underset{j=1, j \neq k}{\overset{m}{\forall}} P(a_j \mid S_i)$, i.e.

$$\forall S_i : \exists a_k \in \Sigma : \underset{j=1, j \neq k}{\overset{m}{\forall}} a_j \text{ it holds that } \left[P(a_k \mid S_i) > P(a_j \mid S_i) \right]$$

If the action a_j occurs at a state $S_i \in Q$ then the transition occurs to the same state, i.e., $\delta(S_i, a_j) = S_i$. If the action a_k occurs at a state $S_i \in Q$ then the transition occurs to the next state, i.e., $\delta(S_i, a_k) = S_{i+1}$. The start state and the final state of a FSM are represented by $q_0 = S_1$ and $F = \{S_n\}$ respectively.

The general flow of the algorithm for probabilistic intention recognition using state machines is shown below in the Fig. 3. The flow diagram shows that an intention is recognized if the concerned FSM reaches its final state and it has the highest weight as compared to other FSMs.

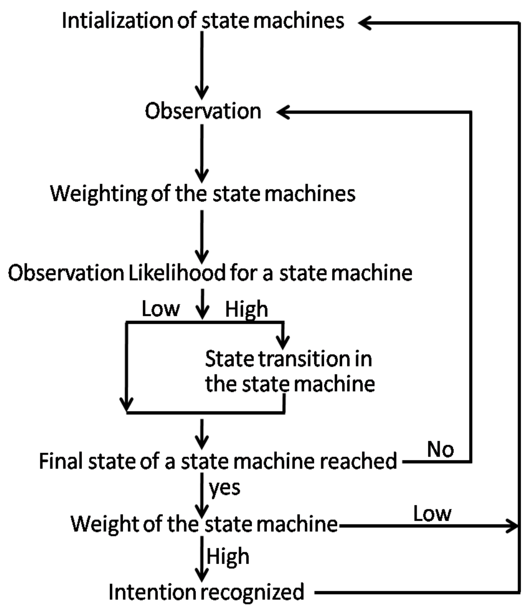

Fig 3. Flow of intention recognition algorithm

III. Trigger State Determination

Proactive and in time reactions are important for intuitive human-robot interaction. The procedure given in Fig. 4 describes the method for making the reaction as proactive as possible by selecting the earliest possible trigger states of the

```
TS : Trigger State
idx : IndexOf
1–  CREATE FSM_{n+1}
2–  FSM_{n+1}.startState.TS = true
3–  FOR FSM_i    i = 1,...,n
4–    FOR all the j states of FSM_{n+1}
5–      IF( j < FSM_i.size())
6–        IF(Match(FSM_{n+1}.state.at(j), FSM_i.state.at(j)))
7–          incr j
8–        ELSE
9–          FSM_{n+1}.state.at(j).TS = true
10–       END IF
11–       IF( j > idx(FSM_i.TS))
12–         FSM_i.state.at(idx(FSM_i.TS)).TS = false
13–         FSM_i.state.at(j).TS = true
14–         break                    // exit loop at step 4
15–       END IF
16–     ELSE
17–       FSM_i.state.at(FSM_i.size() − 1).TS = true
18–       FSM_{n+1}.state.at(FSM_i.size() − 1).TS = true
19–       break                      // exit loop at step 4
20–     END IF
21–   END FOR                        // Go back to step 4
22–   IF( j == FSM_{n+1}.size() || j == FSM_i.size() )
23–     IF( j == FSM_{n+1}.size() && j == FSM_i.size() )
24–       delete FSM_{n+1}
25–       break                      // exit loop at step 3
26–     END IF
27–     FOR all the k Groups
28–       IF(FSM_i ∈ G_k)
29–         assign FSM_{n+1} to Group G_k
30–         update priors of G_k
31–         break                    // exit loop at step 3
32–       END IF
33–     END FOR                      // Go back to step 27.
34–     IF(FSM_i ∉ G_k)
35–       assign FSM_i and FSM_{n+1} to G_{k+1}
36–       assign priors to the G_{k+1}
37–       break                      // exit loop at step 3
38–     END IF
39–   END IF
40– END FOR                          // Go back to step 3.
```

Fig 4. Trigger state determination algorithm

FSM_i, $i = 1,...,n$. The *trigger state* is the end state which means the state that finishes the intention recognition process. The FSM_i, $i = 1,...,n$ represent the human intentions. The input to the procedure in Fig. 4 consists of all the previous FSM_i and the newly constructed FSM_{n+1} that may be added to a group of Finite State Machines (FSM_s). The trigger state of newly constructed FSM_{n+1} is updated with respect to all the previously existing FSM_i. The output of the procedure corresponds to the possible new groups of FSM_s and the update of the trigger state of the FSM_i with in the exiting groups and with out the groups.

In this procedure, the newly constructed FSM_{n+1} is added by comparing with all the previously existing FSM_i. Initially the start state of FSM_{n+1} is considered as the trigger state (Line 2). The comparison between FSM_{n+1} and FSM_i is performed by comparing the transition conditions of the states of both the FSM_s, i.e., FSM_{n+1} and FSM_i, in a sequence (Lines 2-20). If the during the comparison of FSM_{n+1} and FSM_i (Line 5) the state index j of FSM_{n+1} increases the size of FSM_i then update of trigger state is performed for FSM_{n+1} and FSM_i (Lines 17-18). If FSM_i already belongs to a group G_k, i.e., $FSM_s \in G_k$ then FSM_{n+1} is added to that group (Lines 28-31). The intention prior values of the FSM_s belonging to the group G_k are updated. The *intention priors* correspond to the prior probabilities of concerning $FSM_s \in G_k$. If FSM_i does not belong to a group then a new group is created (Lines 34-38). If a mismatch occurs (Line 6) then the trigger state assignment is performed for the FSM_{n+1} (Line 9). The trigger state of the FSM_i is also updated if necessary (Lines 11-15). If the state index j corresponding to the last successful comparison between FSM_i and FSM_{n+1} is greater than the current trigger state index of FSM_i (Line 11) then the update is performed for the trigger state for FSM_i. Otherwise no update of trigger state is performed for the FSM_i. The trigger state always moves toward the actual end state of FSM_i during the process of comparison. The lines 23-26 corresponds to the situation if FSM_{n+1} and FSM_i are same and are of same size then FSM_{n+1} is removed and procedure exits (Line 25). The lines 22, 27-38 corresponds to the situation if FSM_{n+1} and FSM_i are same and one is of bigger in size form the other. The procedure stops if initial part of FSM_i matches to complete FSM_{n+1} or vice versa and if there exists a group G_k such that $FSM_i \in G_k$ (Line 28) then FSM_{n+1} is assigned to that group G_k. Otherwise a new group G_{k+1} is created and FSM_{n+1} and FSM_i are assigned to that group.

The matching of initial part of FSM_i to FSM_{n+1} means that the sequence from the start state of FSM_i to some intermediate state of FSM_i matches to FSM_{n+1} from the start state to the end state with respect to the state transition conditions. The matching can also occur in the reverse manner, i.e., initial part of FSM_{n+1} matches to a complete FSM_i.

A group G of FSM_s is only created if there is a FSM_i and FSM_{n+1} such that they exactly match with each other and one is bigger than the other and already no group exists (Lines 34-38). In case, if a group already exists then FSM_{n+1} is simply added to that group (Lines 28-32). If a group is constructed then a common trigger state is nominated for the group that is the actual end state of the smallest FSM in the group. If that trigger state is reached then the intention selection in the group is performed depending on the intention priors of the state machines in the group, $FSM_s \in G_k$. Initially when the intention priors are uniform (Line 36) then the intention selection is performed randomly and the switch between the different intentions (represented by different $FSM_s \in G_k$) in the group is performed by the human interruption.

After an intention is recognized the intention priors are updated accordingly, e.g., if we suppose there are three FSM_s in a group G_k that initially have the uniform priors of 1/3 then if the intention concerning SM_1 out of three in the group is recognized then the priors will be updates as given in Fig. 5.

$$SM_1 = \frac{1}{3}, \quad SM_2 = \frac{1}{3}, \quad SM_3 = \frac{1}{3}$$

Updation:

$$SM_1 = \frac{1}{3} + \frac{1}{3} \quad = \frac{2}{3}$$

Normalization:

$$\frac{2}{3} + \frac{1}{3} + \frac{1}{3} \quad = \quad \frac{4}{3} \quad = \quad 1.3333$$

$$SM_1 = \ \frac{4}{3}\Big/1.3333, \quad SM_2 = \frac{1}{3}\Big/1.3333, \quad SM_3 = \frac{1}{3}\Big/1.3333$$

$$SM_1 = \frac{1}{2}, \quad SM_2 = \frac{1}{4}, \quad SM_3 = \frac{1}{4}$$

Fig 5. Trigger state determination algorithm

The time complexity of this algorithm is given below where n

$$O(n) \quad = \quad m \geq x \leq n.(m-1)$$

is the total number of the existing state machines and m is the number of the states of the newly constructed state machine FSM_{n+1}.

The best case occurs if FSM_{n+1} already exist or it belongs to some already existing group. The normal case involves FSM_{n+1} that does not belong to any existing group. The worst case occurs if all the state machines FSM_i have the same initial part as the FSM_{n+1} till the second last state of FSM_{n+1}.

The Fig. 6 shows the recognition of the intention depending upon the priors of the FSM_s in case if common end state relating to a group is reached.

IV. Online Update of Local Transition Weight

The *local transition weights* correspond to the weights assigned to the transition conditions in a FSM_i. It is a step forward towards the granulation of intention weights. The local transition weights help the robot to take the premature intuitive decision for intention recognition.

The local transition weights are calculated for the transition conditions that are common among different FSM_s. Every unique transition condition is given the maximum weight, i.e., 1 that is not common among a group of FSM_s. Here the common transition conditions mean the conditions common with respect to the observation's specification and the state's place, i.e., the states are equally apart from the start state and the previous transition conditions, if exist, are the same. These FSM_s are grouped together based on the common conditions. The group FSM_s is not the same as described earlier in Fig. 4 (Lines 28-39). The characteristics of common and unique transition conditions are explained through the Fig. 7. In the Fig. 7 a_i, b_i and c_i represent the transition conditions.

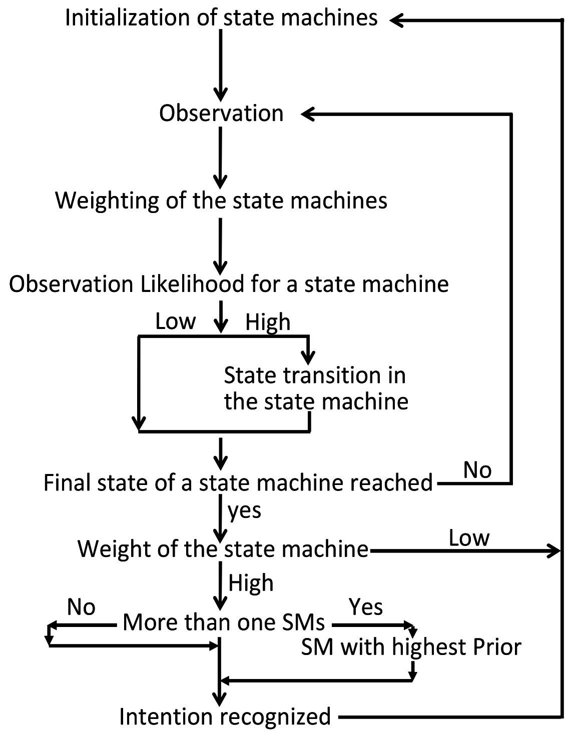

Fig 6. Updated Flow of Intention recognition algorithm

For example the observation of a specific change in distance between the objects may lead to a specific pattern (intention) in the human's mind.

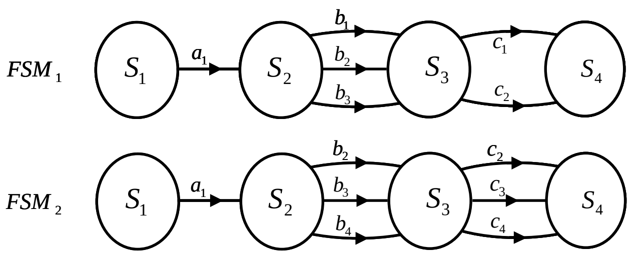

Fig 7. SM_1 and SM_2 representing the common transition conditions

The unique transition conditions a_1, a_2, b_1, b_4, c_1, c_3, c_4 get the weight of 1 and the transition condition b_2, b_3 get initially the uniformly distributed weights among the common transition condition, i.e., 0.5. The weights of b_2 and b_3 are updated with the recognition of the intention represented by the FSM_s relating to b_2 and b_3.

At the construction time of the *FSM* different probabilities are assigned to a transition condition between the neighbouring states. The transition condition that is highly likely to occur at the state and leads to the next state gets the highest transition probability. This highest probability is used as a threshold for the state transition from the state to the next state.

In case if there exist more than one equally highly likely transition conditions then these transition conditions are equally assigned high transition probability, i.e., if any of these observations occur then the state transition occurs.

There may be the case as shown for FSM_1 and FSM_2 in the Fig. 7 that some of the highly likely transition conditions are common among different FSM_s. These common transition conditions among the FSM_s, in a group, are initially assigned the uniform local transition weights. The update of the weights is performed by the addition of $1/|FSM_s|$ to the weight of transition condition that belongs to the *FSM* representing the recognized intention and then doing the normalization as shown in Fig. 5. The $|FSM|$ represent the number of FSM_s having the common transition condition in a group.

Since for a transition to occur between the states the observed transition condition should have the transition value greater or equal to the threshold value. The common transition condition that was earlier unique and had the maximum local transition weight and had the maximum observation probability could trigger the transition. However, as a common transition condition the local transition weight is reduced to $1/|FSM_s|$. Thus the assigned maximum transition probability of the common observation, multiplied by the local transition weight can not trigger the transition to the next state. It will take very long that the weight of the common transition is updated very near to one and the weight of the other related common transition observations near to zero. Then that common transition condition with updated weight near to 1, multiplied by the transition probability may cause the transition.

For the purpose of the faster increment in the update of transition condition's weight an adaption factor θ is introduced. That is also multiplied by the local transition weight and transition probability to calculate the transition value. The adaption factor θ may be changed in order to adjust the adaption rate. The adaption factor used for different no of FSM_s is given below

$FSM = 2 \qquad \theta = 1.3$

$FSM > 3 \qquad \theta = \frac{|SM|}{2}, \frac{|SM|}{3}, \frac{|SM|}{4}, \ldots$

The local transition weights are further explained by an example using two FSM_s. These two FSM_s have one common transition condition as shown in the Fig. 8 given below

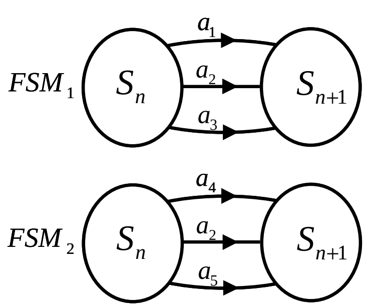

Fig 8. FSM_s with common transition condition

There exist three transition conditions a_1, a_2, and a_3 in FSM_1. FSM_2 has transition conditions a_2, a_4, and a_5 for the states S_n to S_{n+1}. The transition condition a_2 is common among the FSM_s. Therefore, initially the transition condition a_2 in both the FSM_s get the uniform local intention weight, i.e., 0.5 and all the other transition condition get the highest weight, i.e., 1. Now whenever the observation of a_1, a_3, a_4, a_5 occurs then a transition occurs from the state S_n to S_{n+1}.

As the observations that are highly likely for a state are given the high observation probability for that state. Therefore the transition condition a_1, a_2, a_3, a_4, a_5 have the highest transition probability. The adaption factor for all the unique transition conditions is 1. Therefore calculating the transition value for the transition condition a_1, a_3, a_4, a_5 will give the value equal to the transition threshold for the concerned states. The transition threshold is equal to the highest transition probability between the consecutive states. The transition due to a transition condition only occurs if the calculated transition value for that transition condition is greater or equal to the transition threshold for the state. In case of transition condition a_2, no state transition will occur in both the state machines FSM_s.

Although the transition probability initially assigned to a_2 in both the FSM_s is equal to the threshold value. But the local transition weight is 0.5 and the adaption factor is 1.3. Thus the calculated transition value will be less than the transition threshold value as shown below

$$
\begin{aligned}
&= 0.5 \times transition\_value \times 1.3 \\
&= 0.65 \times transition\_value \\
&< threshold\_value
\end{aligned}
$$

If the human performs an action and the observation relates to one of the unique transition condition of a *FSM* then the local transition weights of all the common transition conditions are increased for concerned *FSM* and for the other FSM_s are decreased.

$s_1 = SM_{1:a2} \quad s_2 = SM_{2:a2} \quad \theta = adaption\ \ factor = 1.3$

$Transition\_value\ = \pi$

Step 1:

Updation

$s_1 = 0.5 + 0.5, \qquad s_2 = 0.5$

$$T = \sum_{i=1}^{2} s_i = 1 + 0.5 = 1.5$$

Normalization

$$s_1 = \frac{s_1}{T} = 0.66667$$

$$s_2 = \frac{s_2}{T} = 0.33333$$

$$\theta \times \arg\max(s_1, s_2) \quad = \quad 1.3 \times 0.66667 = 0.86667 \times \pi \ < \ \pi$$

Step 2:

Updation

$s_1 = 0.66667 + 0.5, \qquad s_2 = 0.33333$

$$T = \sum_{i=1}^{2} s_i = 1.667 + 0.33333 = 1.5$$

Normalization

$$s_1 = \frac{s_1}{T} = 0.77780$$

$$s_2 = \frac{s_2}{T} = 0.22222$$

$$\theta \times \arg\max(s_1, s_2) \quad = \quad 1.3 \times 0.77780 = 1.0111 \times \pi \quad \geq \pi$$

Fig 9. Update and Normalization of local transition weights for common transition conditions using the adaption factor θ

m_i Local transition weight $i = 1, 2, 3, ..., n.$

n Number of FSM_s with common condition

$m_i = \frac{1}{n}$ Initially

Step 1:

Weight updation

$$m_1 = \frac{1}{n} + \frac{1}{n} = \frac{2}{n}$$

Normalization

$$\sum_{i=1}^{n} m_i = \frac{2}{n} + (n-1) \times \frac{1}{n} = \frac{n+1}{n}$$

$$m_1 = \frac{2}{n+1} \;, m_2 = \frac{1}{n+1}, m_3 = \frac{1}{n+1}, ..., m_n = \frac{1}{n+1}$$

Step 2:

Weight updation

$$m_1 = \frac{2}{n+1} + \frac{1}{n} = \frac{3n+1}{n(n+1)}$$

Normalization

$$\sum_{i=1}^{n} m_i = \frac{3n+1}{n(n+1)} + (n-1) \times \frac{1}{n+1} = \frac{n+1}{n}$$

$$m_1 = \frac{3n+1}{(n+1)^2} \;, m_2 = \frac{n}{(n+1)^2}, ..., m_n = \frac{n}{(n+1)^2}$$

Step 3:

Weight updation

$$m_1 = \frac{3n+1}{(n+1)^2} + \frac{1}{n} = \frac{4n^2+3n+1}{n(n+1)^2}$$

Normalization

$$\sum_{i=1}^{n} m_i = \frac{4n^2+3n+1}{n(n+1)^2} + (n-1) \times \frac{n}{(n+1)^2}$$

$$= \frac{n^3+3n^2+3n+1}{n(n+1)^2}$$

$$m_1 = \frac{4n^2+3n+1}{n^3+3n^2+3n+1}$$

$$m_2 = \frac{n^2}{n^3+3n^2+3n+1}, ..., m_n = \frac{n^2}{n^3+3n^2+3n+1}$$

Step 4:

$$m_1 = \frac{5n^3+6n^2+4n+1}{n^4+4n^3+6n^2+4n+1}$$

$$m_2 ... m_n = \frac{n^3}{n^4+4n^3+6n^2+4n+1}$$

Step 5:

$$m_1 = \frac{6n^4+10n^3+10n^2+5n+1}{n^5+5n^4+10n^3+10n^2+5n+1}$$

$$m_2 ... m_n = \frac{n^4}{n^5+5n^4+10n^3+10n^2+5n+1}$$

Step 6:

$$m_1 = \frac{7n^5+15n^4+20n^3+15n^2+6n+1}{n^6+6n^5+15n^4+20n^3+15n^2+6n+1}$$

$$m_2, ..., m_n = \frac{n^5}{n^6+6n^5+15n^4+20n^3+15n^2+6n+1}$$

$$\vdots$$

Figure 10: Calculated Local Transition weights

Now if a unique observation relating FSM_1 at state S_n occurs and intention regarding FSM_1 is recognized then the local transition weight of a_2 in FSM_1 is increased and local intention weight of a_2 in SM_2 is decreased. The update is performed by the addition of 1/| FSM_s|, i.e., the average value of the numbers of FSM_s having the common transition condition.

In the above described example there are two FSM_s having one common transition condition. The maximum local transition weight is multiplied with the adaption factor to calculate the transition value as shown in Fig. 9.

As the intention related FSM_1 is recognized thus the local intention weight of a_2 is increased by 0.5. The adaption factor θ increases the local transition weight to the extent that a common transition condition in a specific *SM* is triggered as shown in the calculation, given in Fig. 9.

For that intention to be recognized the human produces the unique transition condition relating to the concerned *SM*. The common transition condition causes the transition between the states for a specific *FSM* that represent the recognized human intention.

If θ is selected as |*FSM*|/2 then the adaption rate for a common transition condition a_{ij} (at i^th state of j^th finite state machine) of FSM_j becomes 2 for |*FSM*| > 3. The adaption rate of 2 means that if an intention represented by FSM_j is recognized 2 times consecutively with respect to other FSM_s in a group having the common transition conditions. Then the local transition weight of a_{ij} of FSM_j is increased and the weights of other related common transition conditions in the group FSM_s are decreased. The two times consecutive increments of local transition weight of a_{ij} and the scaling performed with |FSM_s|/2 causes the state transition due to a_{ij} for |FSM_s| > 3. If |FSM_s| ≤ 3 then three times consecutive increments in the local transition weight of a_{ij} is required to trigger the a_{ij} state transition. Similarly, if θ = |FSM_s|/3 then the specific increment in the local transition weight requires 3 steps for |FSM_s| > 7. In case if 5 ≤ |FSM_s| ≤ 7 then 4 steps are required. The Table-1 given below describes the number of steps required with respect to the |FSM_s| and θ.

θ	No of Steps	|SM|
| FSM_s |/2	2	> 3
| FSM_s|/2	3	= 3
| FSM_s |/3	3	> 7
| FSM_s |/3	4	≥ 5
| FSM_s |/3	5	≥ 3
| FSM_s |/4	4	> 11
| FSM_s |/4	5	≥ 8
| FSM_s |/4	6	≥ 6
⋮	⋮	⋮

Table-1: Description of θ with respect to |*FSM*| and no of steps

The local transition weights are calculated in terms of 1/|FSM_s| as shown in the Fig. 10. The calculation is so performed that the local transition weight m_1 is increased at each step by 1/*n* = 1/| FSM_s|. At each step m_1 is updated (increased by 1/*n*) and then normalized. The six step update and normalization is performed for m_n local transition weights in Fig. 10. Thus m_1 increases and $m_{2...n}$ decrease. Therefore it can be easily checked by multiplying the θ with m_1 at different steps that how many consecutive steps (weight increments) are required for increment of m_1 such that m_1 can cause state transition, e.g., if we take θ = $|SM|/2$ and $|SM|$= 3 m_1 can cause state transition, results are shown in the Fig. 11.

It is also mentioned above in the Table-1 row 2 that at step 3 the local transition weight (updated and normalized) multiplied by θ causes the state transition. That value multiplied with the transition probability (a_j) will not decrease the calculated transition threshold value and will cause the state transition.

$$n = |FSM_s| = 3$$

$$Step\ 1: m_1 = \frac{2}{n+1} \times \frac{n}{2} = \frac{2}{4} \times 1.5 = 0.75 < 1$$

$$Step\ 2: m_1 = \frac{3n+1}{(n+1)^2} \times \frac{n}{2} = \frac{3\times 3+1}{(3+1)^2} \times 1.5 = 0.93750 < 1$$

$$Step\ 3: m_1 = \frac{4n^2+3n+1}{n^3+3n^2+3n+1} \times \frac{n}{2} = \frac{36+9+1}{64} \times 1.5$$
$$= 1.0781 > 1$$

Fig 11.Consecutive increment of a local transition weight

V. EXPERIMENTS

The experiments have been performed with a robotic arm of 6 degrees of freedom. The human and the robot interact in a human-robot interaction workspace shown in Fig. 12. The work space consists of a table with objects and buttons on the table along with the robotic arm. The video data is captured with an over head FireWire digital camera with the frame size of 640 x 480. The camera provides video data at the speed of 30 frames / sec. Human-robot interaction and image analysis is implemented using C++. The robot reactions are realized using the V++ for the robotic arm. The robot is communicated the cooperative instructions using the TCP/IP connection for assigning different operation, e.g., pick, place and move to a certain location, etc. Common Skin detection, Edge detection algorithms and Fourier descriptors are used for the image analysis.

Figure 12: Human-robot interaction workspace

The buttons on the table include Stop (S), Learn (L), Pause (PA), Play (PL), and Reset (R) as shown in Figs 12, 13, 14, and 15. These buttons are used by the human to interface with the robot while human-robot interaction. If the human wants to teach the robot about his intention then the human puts the hand on the L button. Afterwards the human perform the intended task. The S button is used by the human if the human wants to stop the robot from performing a task and undo the current robot action. The robot temporarily stops its activity if the PA button is used. If the PL button is used then the robot starts recognizing the known intentions and after recognizing an intention the robot reacts accordingly. The R button is used to remove all the known intentions that are stored as FSM_s.

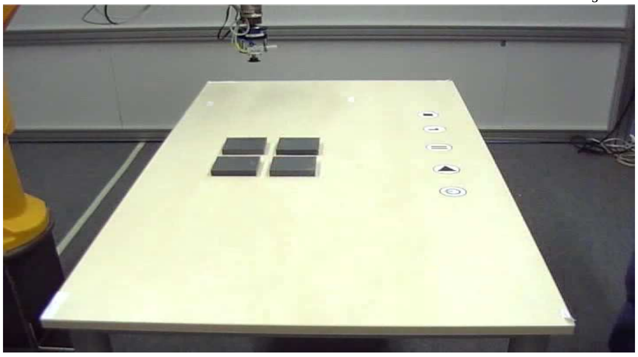

Fig 13. Intention for placing the boxes in a square pattern

The perception of human intention is performed based on the Case 3 discussed in [2], i.e., the human actions and intention is recognized from the scene changes occurred due to the human action. For performing the experiment regarding the arrangements of objects on the table, different human intentions are taught to the robot [2]. The two taught human intentions are shown in the Fig. 13 and 14.

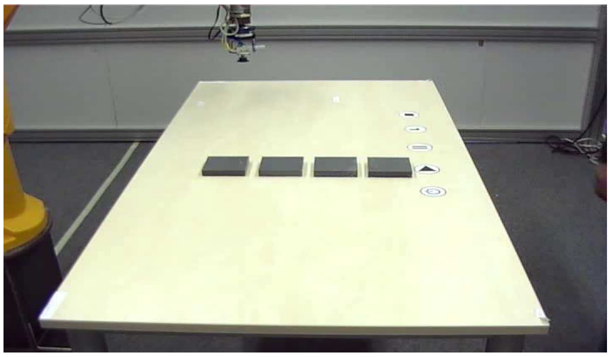

Fig 14. Intention for placing the boxes in longitudinal pattern

The Fig. 15 shows the similarity of the situation (ellipse) for which the robot needs to decide for premature action selection. First the intentions relating to Fig. 13 and 14 are taught to the robot.

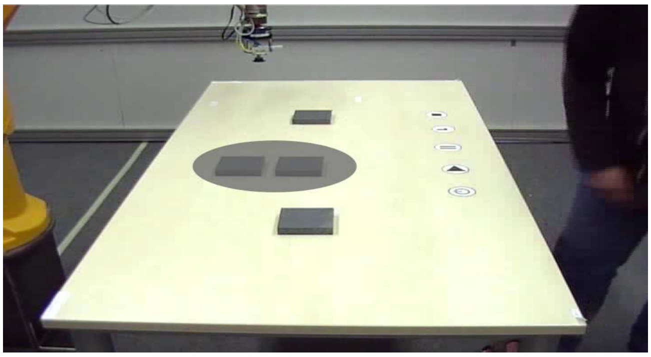

Figure 15: Human robot interaction workspace

Then the robot is presented the situation shown in Fig. 15. The robot can not decide how to react in the situation shown in Fig. 15. The robot waits for the human to disambiguate the situation. Now, if the human performs the action regarding to one of the intentions as shown in Fig. 13 and 14 then the local transition weight of the common transition condition in concerned *FSM* is increased and for the other *FSM* is decreased. Initially, the ambiguous case as shown in Fig. 15, if a task is disambiguated consecutively two times and third time the robot is faced with the ambiguous situation then the robot reacts accordingly, i.e., the robot performs the most likely human intended task in that situation.

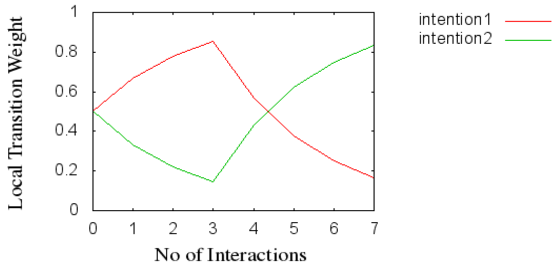

Fig 16. Local transition weights with out adaption factor

The graph in Fig. 16 corresponds to the local transition weights in two FSM_s with one common transition condition as shown in Fig. 13 and 14. Initially, at Step 0 the local transition weights are uniform, i.e., zero for both the common transition conditions. The local transition weight represented by red line represent the transition condition whose concerning intention is selected consecutively three times. Thus the red line rises and green line falls. At the Step 2, the local transition weight (red line) is less than 1 as shown in Fig. 16.

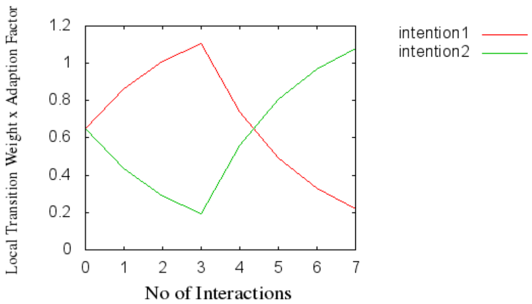

Fig 17. Local transition weights with adaption factor

The local transition weight scaled with adaption factor 1.3 reaches the value 1 at Step 2, as shown in Fig. 17 and causes the state transition. Form Step 3-7 the local transition weight of transition condition (green line) is increased due to the consecutive selection of the concerning intention as shown in Fig. 16 and 17. At Step 7, the local transition weight (green line) in combination with adaption factor can cause transition.

Similarly in the case of trigger state determination and update the premature intention recognition is performed with the help of priors. The robot reacts according to the intention of highest prior *FSM* in the group. If the human intends an *FSM* with lower prior then the robot switches to the next intention (*FSM*) with the next highest prior. The priors of FSM_s are updated such that the prior of intention (*FSM*) that is successfully applied is increased and the priors of the others are decreased. The priors of two FSM_s in a group are shown in the Fig. 18. The graph in Fig. 18 represents that for first 11 interactions an intention is selected consecutively and for the rest of 9 interactions the other intention is selected consecutively.

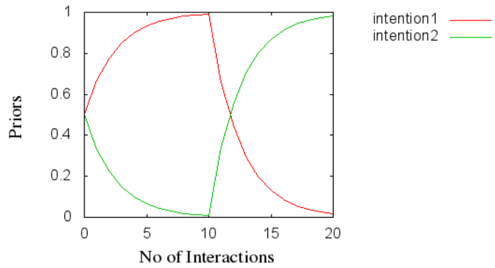

Fig 18. Priors alternating due to the intention switch after 10th interaction.

VI. Conclusions

In this paper we presented a probabilistic proactive approach for the intuitive human-robot interaction in the ambiguous situation. Two cases were discussed for proactive robot response for intuitive human-robot interaction. For making the robot interactions as proactive as possible, trigger state selection algorithm is discussed that describes how the trigger states are selected in case of similar state sequence of different FSM_s. In the second case the proactive nature of human-robot interaction is discussed at lower level, i.e., the ambiguous (leading to two or more different human intentions) human action performed by the interacting human is probabilistically handled for proactive human-robot interaction.

References

[1] Armano, G., Cherchi, G., and Vargiu, E. *An Agent Architecture for Planning in a Dynamic Environment.* In Proceedings of 7th Congress of the Italian Association for Artificial Intelligence on Advances in Artificial Intelligence (AI*IA 2001), 2001.

[2] Awais, M., and Henrich, D. *Online Intention learning for human-robot interaction by scene observation* (submitted to HRI 2012)

[3] Awais, M., and Henrich, D. 2010. *Human-Robot Collaboration by Intention recognition using Probabilistic State Machines*, 19th IEEE International Workshop on Robotics in Alpe-Adria-Danube-Region.

[4] Cesta, A., Cortellessa, G., Pecora, F., and Rasconi, R. *Supporting Interaction in the ROBOCARE Intelligent Assistive Environment*, in Proceedings of AAAI Spring Symposium on Interaction Challenges for Intelligent Assistants 2007, pp. 18-25.

[5] Gunderson, J., Schreiber, G., Albu-Schaffer, A., and Hirzinger, G. Programming by Touch: The Different Way of Human-Robot Interaction. Industrial Electronics, IEEE transactions on, 50(4): 659−666, 2003.

[6] Kurazume, R., Omasa, H., Uchida, S., Taniguchi, R., and Hasegawa, T. *Embodied Proactive Human Interface "PICO-2". In Proceedings of the 18th International Conference on Pattern Recognition (ICPR' 06),*

[7] Miura, J and Shirai, Y. *Parallel Scheduling of Planning and Action of a Mobile Robot Based on Planning-Action Consistency*. In Proceedings of the IJCAI-99 Workshop on Scheduling meet Real-time Monitoring in a Dynamic and Uncertain World; Stockholm, Sweden, 1999.

[8] Madhava, Krishna., Alami, R., and Simeon, T. *Safe proactive plans and their execution*. Robotics and Autonomous Systems, 54(3):244−255.

[9] Schmid, A., Weede, O., and Worn, H. *Proactive Robot Task Selection Given a Human Intention Estimate*. Robot and Human interactive Communication, RO-MAN. 16th IEEE International Symposium.

[10] Wada, K. and Shibata, T. *Living With Seal Robots Sociopsychological and Physiological Influences on the Elderly at a Care House*, IEEE Transactions on Robotics, Vol. 23, No. 5, October 2007.