Authors´ accepted version. https://doi.org/10.1145/3453892.3453895

Probabilistic Combination of Heuristic Behaviors for Shared Assistive Robot Control

Felix Goldau
felix.goldau@dfki.de
German Research Center for Artificial Intelligence (DFKI)
Bremen, Germany

Udo Frese
udo.frese@dfki.de
German Research Center for Artificial Intelligence (DFKI)
Bremen, Germany

Figure 1: User Perspective of the Adaptive DoF Control, including a robot arm with wrist-mounted camera (left) and the user interface as shown on the smart glasses (right)

ABSTRACT

Technology in general is developed to improve the lives of their users, often by allowing them to handle individual struggles. Assistive robotics takes this concept to its extreme by (re-) enabling users to physically interact with their environment in their daily life. For a successful utilization however, a user interface is required that allows for easy and quick interaction. Based on the promising concept of Adaptive Degree of Freedom (DoF) Control, this paper presents a novel heuristic implementation of the underlying principle, without relying on learning user actions.

To achieve this, a mixture distribution is obtained which expresses how likely the user wants which motion. Here, every mode of the mixture represents a heuristic behavior. Each such behavior defines its own distribution of motion, as well as a weight indicating how likely it is in the current situation. The best fitting DoF is obtained from this mixture and offered to the user with an interface.

This general-purpose control method has been tested in a small technical study, the results of which show its general viability, promising chances for a significant reduction of mode changes, as well as very good quantitative feedback by the users.

PETRA '24, June 26–28, 2024, Crete, Greece

ACM ISBN 979-8-4007-1760-4/24/06
https://doi.org/10.1145/3652037.3652071

CCS CONCEPTS

• **Human-centered computing** → *Accessibility technologies*; Empirical studies in accessibility; User centered design.

KEYWORDS

Adaptive DoF Control, Latent Action Space, Assistive Robotics, Social Robotics, Shared Robot Control

ACM Reference Format:
Felix Goldau and Udo Frese. 2024. Probabilistic Combination of Heuristic Behaviors for Shared Assistive Robot Control. In *The PErvasive Technologies Related to Assistive Environments Conference (PETRA '24), June 26–28, 2024, Crete, Greece*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3652037.3652071

1 INTRODUCTION

Assistive robot arms help users with physical impairments who cannot use their own arms and hands to perform activities of daily living (ADLs). However, robot arms are complex devices with usually six degrees of freedom (DoFs) plus at least one for the hand. This makes designing an accessible and effective user interface for the robot difficult. On one hand, special input devices with many degrees of freedom, e.g. a 3D mouse, exist, but they require considerable dexterity which most people of the target group cannot exercise. On the other hand, input devices specifically tailored for this application, e.g. a head motion based joystick, only offer to input one or two degrees of freedom. Overall the challenge is the mapping of these two input DoFs onto the seven output DoFs.

The standard solution is a menu to switch between different assignments of cardinal DoFs, so-called modes, e.g. X, Y, Z, roll, pitch, yaw, gripper. As the required motions in ADLs are usually not aligned with a cardinal coordinate system, many mode switches are required slowing down the robot's use. For example pouring a glass of water takes $\approx 500\ s$ with ≈ 50 mode switches[7]. Our research investigates the idea that artificial intelligence shall analyse the situation from images of a hand mounted RGBD-camera and suggest a DoF the user will probably want to use. Usually, this will not be a cardinal one. If the computer is right, time is saved and efficiency gained; if the computer is wrong, the user can still manually choose an alternative DoF.

This shared control scheme leaves the user in command and reduces requirements for the artificial intelligence that does not need to be as perfect as in pure autonomous operation. Conceptually this can be viewed as a probabilistic prediction of $p(U_t|Z_{1:t})$: How likely is it, that the user will want the motion U_t, given the situation $Z_{1:t}$ as apparent from sensors such as the camera? With the large success of deep learning, it is tempting to learn $p(U_t|Z_{1:t})$ from recorded data, such as the DORMADL dataset[4]. However, there are some challenges in that, which we will discuss in detail in Section 6. The most notable two are: The necessity to work on image sequences $Z_{1:t}$ instead of single images Z_t, as the hand camera often simply does not see enough of the scene; And the difficulty to incorporate the actually available geometric information from the depth camera and kinematics.

Hence, this paper explores a more analytical alternative. The approach is to define $p(U_t|Z_{1:t})$ as a mixture distribution, where each mixture mode represents an elementary behavior that is heuristically defined and reports its own relevance to the situation, which in turn is used as a weight in the mixture. The behaviors can access the current depth and kinematic data Z_t, as well as the position of tracked objects even if they are currently outside the camera's view. The latter is a way of aggregating information over time, i.e. computing $p(U_t|Z_{1:t})$ instead of $p(U_t|Z_t)$. It is not as general as the $|Z_{1:t}$ notation suggests, but it addresses the most important issue, namely to remember objects outside the field of view.

1.1 Contributions

The contributions of this paper are

- a set of basic behaviors of assistive robots in ADLs,
- in particular an approaching and grasping behavior based on perceived geometry and a generic object segmenter,
- a method for probabilistic combination of behaviors in a shared control setting,
- and a user study showing the validity of the approach as well as opportunities to improve user satisfaction by reducing mode switches.

The paper initially gives a short overview of relevant shared control concepts in assistive robotics, then presents the proposed method and finally reports on the user study.

2 SHARED CONTROL IN ASSISTIVE ROBOTICS

One of the most promising operation concepts for an assistive robot is shared control. Based on the duality of control inputs from a user and software, the concepts generally pairs user input with an automation or similar support by a computer system (e.g. [16] automatically handles end-effector rotation with the user only controlling translational DoFs). This aims to reduce mode switches, mental load, and execution errors by providing targeted assistance during control operations by the user.

As assistive robotics in general aims to (re-) enable users to perform tasks of daily living, the focus of any associated control methodology needs to be the user themselves, therefore requiring interfaces that keep the users in control whilst allowing them to operate the high-DoFs robots[3]. As the user's life should be self-determined and not automated, classic automation-based robot control systems cannot be applied here. Instead, a clear understanding of user intent is vital[1] and needs to be incorporated into the very base of any functional shared control concept. Assuming a known intention, [11] allows to control the arm using a latent action space, where the user's lower dimensional control input results in a high dimensional motion of the robot.

For this work, we will focus on the similar concept of *Adaptive DoF Control* as presented in [5], which follows the idea of adaptively adjusting the DoFs controllable by user interface, dependent on the current situation. Generally, a classic system has a defined set of modes (e.g. [12] with translational, rotational, grasping). The adaptive DoF control effectively reduces the number of modes and adjust them accordingly: By allowing more diverse DoFs, it allows the user to control the robot along directions more appropriate to the current situation, for example by moving the robot along a diagonal instead of the robot-typical jagged motion of going first left, then forward.

3 PROBABILISTIC COMBINATION OF HEURISTIC BEHAVIORS

The field of *Behavior-based Robotics* follows the concept of creating seemingly complex interactions based on minimal and very simple sensor-driven actions (compare [2]). Generally speaking, this is implemented by direct linking of sensor input to specific action, as for example rotating to the right if light was detected on this side. The most prominent examples of this are the tortoise robots by W. Grey Walter[17]. Such behavior-based robots are often intentionally compared to biological systems, as they can quickly react to new sensory input.

Instead of directly generating actions, as in behavior-based robotics, the robot control presented in this work follows a different, but correlated approach: Use statistical features of a set of preferably simple behaviors as the suggested DoF in an adaptive DoF Control (see Sec. 2).

The idea behind this control is not to select isolated behaviors, but to instead create a combination of options via DoFs. This allows for more natural and smoother motions, especially in areas of transition, as well as blending of directions.

The associated general control structure is shown in Figure 2 and will be explained in detail during this work.

3.1 Definition of a Behavior

In this work, a behavior b describes a simple action from a finite set of actions (such as *Lifting the End-Effector*). More specific, it

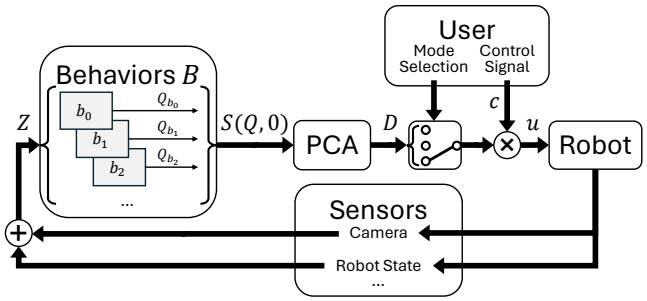

Figure 2: Software control loop. Based on (robot-associated) sensor input Z, the set of behaviors B generate sigma points Q_b, accumulated to a probability distribution S. The user controls the robot along an axis of the principal components of the latter.

is an adaptive multivariate probability distribution describing the likelihood of motion in an n-dimensional direction U_t relative to the end-effector given the situation as observed from the sensor readings $Z_{1:t}$.

$$p_b(U_t|Z_{1:t}) \sim b, \quad U_t \in \mathbb{R}^n \tag{1}$$

The n-dimensional vector U_t describes a velocity, which can be joint angle or Cartesian and include components for the gripper. In our case it consists of a stacked vector with translational, rotational and gripper velocities in end-effector coordinates. Since U_t contains rotational velocity not orientation, there is no problem with singularities or need for quaternions.

In order to ease further processing, each behavior's distribution is defined as a Gaussian $\mathcal{N}(\mu_b, \mathrm{S}_b)$ with mean μ_b and covariance S_b. It is represented by a set Q_b of sigma points, similar to usage in unscented Kalman filters[15] (see Figure 3).

$$\mu_b = \frac{1}{|Q_b|} \cdot \sum_{q \in Q_b} q \tag{2}$$

$$\mathrm{S}_b = \mathrm{S}(Q_b, \mu_b) \tag{3}$$

where $\mathrm{S}(Q, \mu)$ is the covariance of sigma points Q with reference point μ:

$$\mathrm{S}(Q, \mu) = \frac{1}{|Q|} \cdot \sum_{q \in Q} (q - \mu)(q - \mu)^T \tag{4}$$

In this context, each point represents a direction of control which originates at the end-effector's tool center point. The set of points therefore build a distribution with expected value E and covariance Cov conditioned on the behavior's underlying action and the current situation. In addition, each behavior provides a weight ω to represent the likelihood $p(b|s)$ of the underlying action in the current situation.

Using this setup, the simplest behaviors consist of a single sigma point $q_0 \in \mathbb{R}^n$ describing a point distribution ($\mu_b = q_0, \mathrm{S}_b = 0$) of a motion with the direction according to this sigma point. For example, a strict *Forwards* behaviour moves in the direction the end-effector is pointing.

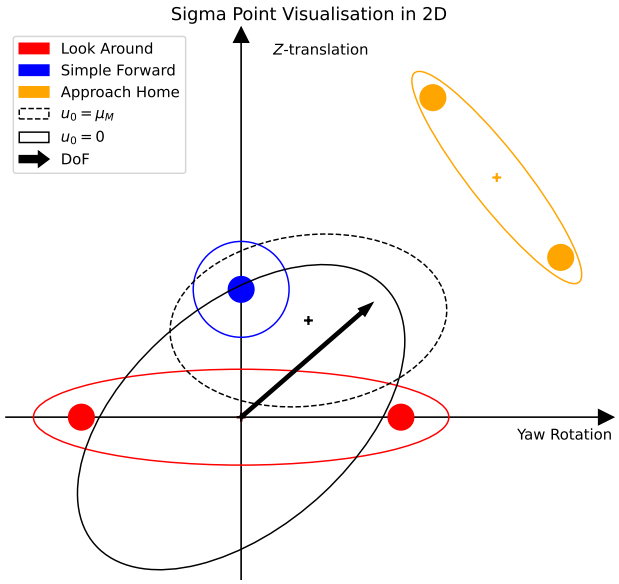

Figure 3: Illustration of the representation of $p(U_t|Z_{1:t})$ as a mixture distribution. The dots are the sigma-points of three behaviors (red, blue, orange), the ellipses show the corresponding mean and covariance. The black dotted ellipse shows the overall mean and covariance of the mixture, the bold ellipse corresponds to S from equation (14), and the black arrow shows the single optimal DoF. Mixture weights are represented by the distance of the sigma points to the origin.

A slightly more complex behavior may, for example, have no preferred sign of the direction (which would be the case for a general DoF) and could therefore represent this by supplying two opposite points $Q_b = \{q_0, -q_0\}$ leading to $\mu_b = 0$ and $\mathrm{S}_b = q_0 {q_0}^T$. An example of this kind of single-DoF behavior is the *Look Around* behavior described in Section 4.1, which can yaw left or right without preference.

More complex behaviors can use arbitrary numbers of sigma points to represent uncertainty in multiple dimensions. These sigma points can also depend on the environment.

If the behavior involves rotation and translation, there are several options: Both can be combined into one sigma-point with a defined ratio. This expresses that the user likely wants a combined motion, e.g. an orbit around an object. Or there can be two sigma-points, one with the rotation and no translation and one with the translation and no rotation. This expresses, that the user likely wants a translation, or a rotation or a combination of both. In theory rotation and translation could also be two different behaviors, expressing, that the user likely wants no combination. However, by the way the following processing is done, this makes no difference.

3.2 Combination of Behaviors as Mixture Distribution

The different behaviors are treated as modes of a mixture distribution with weights, as returned by the behaviors. The rationale

behind this view is that we assume the user wants to follow one of the behaviors, but we don't know which one. As the probability of a behavior becomes larger when its better suited to the situation, the mixture distribution has a high chance to reflect the users intent.

$$p(U_t|Z_{1:t}) = \sum_{b\in B} p(b|Z_{1:t}) \cdot p(U_t|Z_{1:t}, b) \tag{5}$$

$$= \frac{1}{\sum\limits_{b\in B} \omega_b} \sum_{b\in B} \omega_b \cdot p_b(U_t|Z_{1:t}) \tag{6}$$

$$= \frac{1}{\sum\limits_{b\in B} \omega_b} \sum_{b\in B} \omega_b \cdot \mathcal{N}\left(\mu_b, \mathrm{S}_b\right)(U_t) \tag{7}$$

Therefore, we can calculate the expected value E_M and covariance Cov_M of the resulting mixture distribution directly from the sigma points Q_b of all behaviors $b \in B$ as

$$\mathrm{E}_M = \frac{1}{\sum\limits_{b\in B} \omega_b} \cdot \sum_{b\in B} \omega_b \cdot \mu_b \tag{8}$$

$$\mathrm{Cov}_M = \frac{1}{\sum\limits_{b\in B} \omega_b} \cdot \sum_{b\in B} \omega_b \cdot \mathrm{S}(Q_b, \mathrm{E}_M) \tag{9}$$

This is shown in Figure 3 as a black dotted ellipse.

3.3 Choosing the most suitable DoFs

To be usable for the adaptive DoF control, a matrix $D \in \mathbb{R}^{nxm}$ of individual DoFs needs to be generated from the mixture distribution, such that an m-dimensional user control signal $c \in \mathbb{R}^m$ can be mapped to a final robot motion u:

$$u = D \cdot c + u_0 \tag{10}$$

For this, the user input axis c_j controls motion along the DoF represented by column j of matrix D.

Similar to [5], our DoF-mapping D has more rows n than columns m, hence not every u can be obtained by an appropriate c. However,

$$c = D^+(u - u_0), \tag{11}$$

with D^+ as the Moore-Penrose-inverse of D gives the input c that produces a robot motion $Dc + u_0$ as close to u as possible.

Substituting (11) into (10), we want to minimize the expected squared error $\mathrm{E}(\|e\|^2)$ of this control

$$e = u - D \cdot D^+(u - u_0) + u_0. \tag{12}$$

The problem can also be viewed as minimizing the expected squared distance from the distribution to the m-dimensional subspace $\{D \cdot c + u_0 | c \in \mathbf{R}^m\}$. Typically, it is solved using a Principal Component Analysis (PCA)[13] with the optimum $u_0 = \mu_M$ and D consisting column-wise of the m eigenvectors with the largest eigenvalues.

$$\hat{D} = \mathrm{eigen}_{1:m}\left(\mathrm{S}_M\right), \quad u_0 = \mu_M \tag{13}$$

If we were to apply this to an adaptive control, an input of zero (i.e. a non-action of the user) would, instead of a standstill, result in a motion of the robot according to the expected value $u_0 = \mu_M$.

This is neither user-friendly, nor safe, so we change the optimization problem to enforce $u_0 = 0$. For that $\mu_M {\mu_M}^T$ needs to be added to the covariance S_M before calculating the eigendecomposition.

$$D = \mathrm{eigen}_{1:m}\left(S\right), \quad u_0 = 0, \quad S = \mathrm{S}_M + \mu_M {\mu_M}^T \tag{14}$$

The resulting distribution is shown in Figure 3 by the black ellipse, with the black arrow representing the eigenvector with the largest associated eigenvalue.

3.4 Mathematical Derivation

This section derives (14). The PCA provides the optimal solution (13) for an optimally chosen u_0. An arbitrary centered m-dimensional subspace is fitted to the distribution. This is well known[13].

As we defined to enforce $u_0 = 0$ to ensure a stationary robot when no user input is given, we modify the derivation from [13] for a zero-centered subspace. Let U be an $\mathbb{R}^n$ random variable. In our system $\mathrm{E}(U) = \mu_M$ and $\mathrm{Cov}(U) = \mathrm{S}_m$, but we will keep the derivation general. Let D parametrize the desired m-dimensional subspace $\{Dc | c \in \mathbb{R}^m\}$, where the columns of $D \in \mathbb{R}^{n\times m}$ span the subspace and D is orthonormal ($D^T D = I$). In [13] there is an additional center θ, which is fixed to 0 in our case.

Since D is orthonormal, $DD^+U = DD^TU$ is the closest point to U on the subspace and BU with $B = I - DD^T$ is the corresponding error vector e. Our optimization problem is

$$J(D) = E\left(\|BU\|^2\right) = \mathrm{E}\left(U^T B^T BU\right) \tag{15}$$

$$D = \arg \min_{D\in\mathbb{R}^{n\times m}, D^TD=I} J(D). \tag{16}$$

The matrix B is symmetric and idempotent ($B^2 = B = B^T$), so

$$J(D) = \mathrm{E}\left(U^T BU\right) = \mathrm{E}\left(U^T (I - DD^T)U\right) \tag{17}$$

$$= \mathrm{E}\left(U^T U\right) - \mathrm{E}\left(U^T DD^T U\right) \tag{18}$$

$$= \mathrm{tr}\left(\mathrm{E}\left(UU^T\right)\right) - \mathrm{tr}\left(D^T\, \mathrm{E}\left(UU^T\right) D\right) \tag{19}$$

$$= \mathrm{tr}(S) - \mathrm{tr}\left(D^T S D\right), \quad S = \mathrm{Cov}(U) + \mathrm{E}(U)\mathrm{E}(U)^T. \tag{20}$$

This is the same expression as in [13, (19), $W = D, X = U$], except that there $S = \mathrm{Cov}(U)$ and here $S = \mathrm{Cov}(U) + \mathrm{E}(U)\mathrm{E}(U)^T$. To conclude, enforcing $u_0 = 0$ leads to an additional term $\mathrm{E}(U)\mathrm{E}(U)^T$ in S. The rest of the proof is the same, deriving that the optimal $D = \mathrm{eigen}_{1:m}(S)$.

4 BEHAVIORS

For our assistive robotic system, we focussed the set of behaviors on a generalized task of interacting with grasped objects as it is a very common use of the arm. In addition, especially the approach and grasping of objects holds good opportunities for assistive support, as grasping objects often requires detailed alignment from the user, graspable objects can be readily identified and defined, and the task has foreseeable goals, instead of, for example, the inverse *Placement* task, which can end nearly arbitrary.

In the following, we will present the seven developed behaviors, six of which as *basic behaviors* in Section 4.1, and one in more detail in Section 4.2. However, all behaviors presented are treated independently and equally, as they are mixed together (see Section 3.3) instead of being selected individually.

4.1 Basic Behaviors

The six basic behaviors follow a very minimalistic structure and are designed to represent some fundamentals of robotic interaction.

We will briefly present each behavior with its sigma points Q_b in end-effector coordinates EE.

In the following, the terms <Rot $(a)^f$> and <Trans $(a)^f$> are used to describe unit rotations or translations along an axis a of a frame f, whereas <Grasp> describes the unit motion vector of a directional movement leading to closing of the fingers. Also, the transformation matrix $T_{B\leftarrow A}$ maps from coordinate system A to B

Please note, that in this framework, motion is described by a vector with components x, y, z, yaw (around x), pitch (around y), roll (around z), gripper. These refer to the relative motion *velocity* in end-effector-coordinates, not to (relative) poses, so the question of avoiding singularities by matrices or quaternions does not arise.

Look Around. In order to reach a target position, it is often necessary to reorient the gripper towards it. This constant behavior describes the necessary rotation solely around the vertical axis, which aims to avoid possible spillage by not changing the end-effector's alignment with the horizon. In addition, this orientation change towards a target improves the situational awareness of the gripper-mounted camera. Without additional information, this behavior provides a DoF, by supplying two opposite sigma points:

$$Q^T_{LookAround} = \begin{pmatrix} \text{Rot}\left(T_{EE\leftarrow Base} \cdot vertical^{Base}\right)^{EE} \\ -\text{Rot}\left(T_{EE\leftarrow Base} \cdot vertical^{Base}\right)^{EE} \end{pmatrix}, \quad \omega = constant. \tag{21}$$

Forward. Once the end-effector is oriented towards the user's target position, a common motion is driving, relative to the end-effector, forwards. For example, when preparing to grasp, users will often initially align with the object and then move in a direct line to reach a grasping pose. This behavior provides a constant, but light-weighted DoF pointing outwards z from the tool center point of the end-effector. This represents the action of continuing motion in the direction the gripper is pointing. The sigma points form a DoF in both directions, with a tendency to go forwards:

$$Q^T_{Forward} = \begin{pmatrix} \text{Trans}\,(z)^{EE} \\ \text{Trans}\,(z)^{EE} \\ -\text{Trans}\,(z)^{EE} \end{pmatrix}, \quad \omega = constant \tag{22}$$

Grasp. Continuing the though process of aiming to grasp an object, this behavior supplies the actual motion of closing the fingers. This behavior is distinct from the others, as it is the only one affecting this dimension. This avoids the accidental opening or closing of the fingers that would otherwise be possible by multidimensional DoFs. Using the attached depth camera, the behavior reacts to the number of close pixels px in between the gripper that suggest the presence of an object.

If an object is already grasped, the behavior will instead suggest opening the gripper upon standstill, with the likelihood increasing based on the time t_s since last movement and the distance d_g travelled since grasping the object.

$$Q^T_{Grasp} = \begin{cases} \left(-\text{Grasp}\right), & \omega = \text{const} \cdot t_s \cdot d_g \quad \text{if object grasped} \\ \left(\text{Grasp}\right), & \omega = \text{const} \cdot px \quad \text{otherwise} \end{cases} \tag{23}$$

Rotate Upright. In most cases, it is desirable to keep the end-effector upright. This simplifies grasping, avoids dropping or spilling of grasped objects, and is often easier for the user to fathom. Based on the current orientation (roll angle r, pitch angle p) of the end-effector, this behavior provides a rotational motion to reorient the gripper to be upright. The yaw angle has no effect on this behavior.

$$Q^T_{RotateUpright} = \left(-\left(\text{Rot}\,(roll)^{EE} \cdot r + \text{Rot}\,(pitch)^{EE} \cdot p\right)\right), \quad \omega = |r| + |p| \tag{24}$$

Liftoff. After grasping or placing an object, there is generally a short phase of retrieval, where either the object is lifted and positioned, or the arm is retracted from the position of the object. In either case, it makes sense to (slightly) lift the end-effector, as well as retrieve it in the general direction of the robots base $v_{EE\leftarrow Base}$ or the user. This should be a safe direction to move in most cases, as it roughly aligns with the user's line of sight and the robot's joints. This behavior diminishes with the time t_g and distance d_g since grasping.

$$Q^T_{Liftoff} = \left(\text{Trans}\,(vertical)^{EE} + v_{EE\leftarrow Base}\right), \quad \omega = \begin{cases} -\text{const} \cdot t_g \cdot d_g & \text{if object grasped} \\ 0 & \text{otherwise} \end{cases} \tag{25}$$

Approach Home. As certain pre-definable positions have recurring meaning, this behavior provides a direct approach towards a home position, weighted by the distance and relative orientation of the gripper. It's individually comprised of a rotation $\lambda\,(home)$ aiming to align the end-effectors z-axis with the target position, and the translational displacement $v_{home\leftarrow EE}$ of the target pose to the end-effector.

For a wheelchair user, one home pose might be above their wheelchair table, so that they can easily retrieve objects to there. For a detailed explanation of λ and ω in this case, please see Sec. 4.2.

$$Q^T_{ApproachHome} = \begin{pmatrix} \lambda\,(home) \\ v_{home\leftarrow EE} \end{pmatrix} \tag{26}$$

4.2 Approach Object Behavior

The *Approach Object* behavior actively scans for graspable objects in range and provides individual behavior distributions for each of them. Individually, each such distribution represents the action of approaching the specific object. However, in combination they provide a distribution of possible directions that best allow to approach the group of objects. This way it actually encapsulates multiple similar behaviors (one for each object stored) which are aggregated in this section. For a detailed analysis of the resulting performance synergy, see Section 4.3.

In general, per detected object o the behavior provides two sigma points $Q_{ApproachObject(o)}$ and corresponding weights ω^r_o, ω^v_o. The sigma points separately describe translation to, and orientation towards, the goal, as it is assumed that users will perform each of them stepwise one after the other (initially rotate towards the object and approach only then):

Authors´ accepted version. https://doi.org/10.1145/3453892.3453895

$$
\begin{aligned}
Q^T_{ApproachObject(o)} &= \begin{pmatrix} \lambda(o) \\ v_{EE \leftarrow o} \end{pmatrix}, \\
\omega^\lambda_o &= \theta_\lambda\left(|v_{EE\leftarrow o}|, |\lambda(o)|, t_o\right) \\
\omega^v_o &= \theta_v\left(|v_{EE\leftarrow o}|, |\lambda(o)|, t_o\right),
\end{aligned} \tag{27}
$$

where the rotational component $\lambda(o)$ is the angle between the forward-pointing z-axis of the end-effector and the vector towards the object and is calculated as

$$
\lambda(o) = \angle\left(\text{Trans}(z)^{EE}, v_{EE\leftarrow o}\right). \tag{28}
$$

Perception. The sensory input of this behavior is structured around the *FastSAM*[18] implementation of the *Segment Anything Model* (SAM)[10]. This neural network model is designed to isolate segments in a color image and can basically be run on arbitrary images. We use the model as-is and actively refrain from doing any model adjustments, as we want the underlying generalistic behavior.

Using the color images generated by the wrist mounted RGBD camera[8] (see Figure 1), the SAM model provides image masks for a fairly large number of possible objects it detects. As not all the generated image segments are valid real-life objects, we post-process each image segment in order to interpret it as an object and generate a reasonable target.

Using the camera-proved depth data, the 3D physical extends of the objects are calculated based on a camera project matrix and the rotated minimal bounding box. In addition, the direct neighboring area around the objects is checked to verify, that the segment protrudes sufficiently from the background to be able to be grasped; In other words, it checks the sides of the objects for chasms that are deep enough for the fingertips. The depth data is also used to calculate the relative object pose.

Based on this data, objects are treated as targets if they are within reach, have grasp-appropriate physical extends, as well as having sufficient chasms to the objects sides for the gripper to protrude during grasping.

Each such object o is persistently stored, so it is remembered even when it falls outside the camera's field of view. The weights ω^λ_o and ω^v_o of an unseen object however decrease over the time t_o since the object was last seen, so to say *forgetting* the object.

Sigma Points. For every object o remembered, a sub-behavior is generated that handles direction orientation $\lambda(o)$ and approach $v_{EE\leftarrow o}$ towards the object.

For this, the weights are regularly updated using the functions $\theta_\lambda\left(|v_{EE\leftarrow o}|, |\lambda(o)|, t_o\right)$ and $\theta_v\left(|v_{EE\leftarrow o}|, |\lambda(o)|, t_o\right)$ respectively. These are simple functions calculating a relative estimate of the likelihood of the user aiming to grasp the object in this current situation. In our setup, they were designed to embrace objects that are close and aligned with the gripper, resulting in a selection dynamic (see Section 4.3). In detail, they are:

$$
\begin{aligned}
&\theta_v\left(|v_{EE\leftarrow o}|, |\lambda(o)|, t_o\right) = \\
&\quad t_o \cdot \left((v_{\max} - |v_{EE\leftarrow o}|) \cdot \left(\frac{1}{2}\left((\cos|\lambda(o)| + 1) \right) \right)^3 \right)
\end{aligned} \tag{29}
$$

$$
\begin{aligned}
&\theta_\lambda\left(|v_{EE\leftarrow o}|, |\lambda(o)|, t_o\right) = \hat{f_\rho} \cdot \theta_v\left(|v_{EE\leftarrow o}|, |\lambda(o)|, t_o\right) \\
&\text{where } \hat{f_\rho} = \begin{cases} 1 & \text{if } |\lambda(o)| > \rho \\ f_\rho & \text{otherwise,} \end{cases}
\end{aligned} \tag{30}
$$

where $v_{\max}$ is the maximum distance reachable by the robot arm, and ρ is a threshold angle, below which the relative likelihood of the rotate-towards sigma point is reduced by a factor of f_ρ. The latter serves to favor the translational component in the final approach rather than optimizing orientation.

The *Approach Home* behavior (see Sec. 4.1) follows the same principle and ω-scaling.

4.3 Synergy in Shared Control

The presented behaviors can obviously not be used as a baseline for automation; The system's information is only very limited with respect to the environment and the intent of the user. In general, there is too much uncertainty for an automation task.

However, as discussed in Section 2, automation should not be the goal of assistive control strategies. Instead, the presented control was developed with the specific user requirements and relies heavily on interaction of the user, who not only selects the current motion to perform, but instead smoothly controls through them.

As each behavior evaluates its own likeliness and adjusts its weight ω accordingly, the weighted combination of the behaviors provides a quickly adapting set of DoFs. From the perspective of the user, the suggested DoFs barely make any decision between options, but instead provides the user with the means of selecting one themselves.

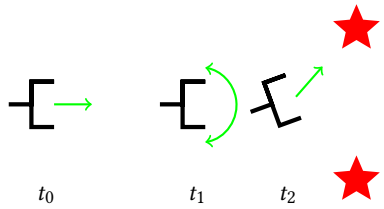

Figure 4: DoF Selection Example: Two star-shaped red targets and a 2D-robot at 3 points in time. For each point in time, a green arrow represents the suggested DoFs that are most likely given the situation.

A simple scenario showcasing these principles can be seen in Figure 4:

Initially, at t_0, the robot is far away from the targets. As the system has no information about which object the user prefers, it simply provides the direction towards the center of the targets, as this will get them closer to their actual goal. This is a direct result of the *Approach object* behavior for each target: The two translational components add up to a clear direction, whereas the rotational components balance out each other and only provide a less likely unsigned DoF.

Assuming the user follows the suggested motion until t_1, the robot has reached a point where the summed translational components have decreased sufficiently, such that they are now smaller than the balanced rotational element. In other words, the system

assumes it's no longer effective to drive forward, but instead orient toward one of the targets. The user must not obey this, but can choose to select a new DoF at any point.

At t_2 the user has obviously continued controlling the robot in a translational manner and has only just started rotating slightly to the left. This orientation towards the leftmost target is however sufficient, such that the system assumes that they prefer this target. As a result, it provides a direct translational approach motion. This is a result of the ω-scaling of θ_λ and θ_v.

In summary, the system relies on the user to make their selection. In the setting described above, if the user had not made any decision to switch, the control would have missed the targets and passed through the center. The systems assist only once the user makes a clear enough decision. The same concept generalizes to an arbitrary number of targets, as well as effects from the other behaviors.

As long as the user input devices supports less input DoFs than the robot has ($m < n$), this method of control cannot be complete (i.e. there are poses the user cannot reach). To solve this, a user interface with a fallback option of controlling in the cardinal DoFs is suggested. This way, the adaptive DoF control only extends existing controls and does not restrict the user's options. See the media attachment for a video example.

Seemingly contradictory, this setup of behaviors does not influence the robot safety. The only but essential rule is that if there is no command from the user, the robot performs no motion. In this setting and with this type of robot, contact with the environment necessarily needs to be possible (be it for object interactions, or tasks like scratching), therefore no specific obstacle avoidance is implemented. This is left to the user.

5 TECHNICAL USER STUDY

To verify the technical usability of the control, we conducted a small user study in a laboratory environment using a Kinova Jaco Gen2 7DoF assistive robot arm[9], an Intel Realsense D435[8] and a ros-based software stack. As this was designed as a proof of technical concept, not an end-user compatibility scoring, we opted to test the control with able-bodied participants.

Based on the scenarios from [4], we selected a simplified supermarket shelf scenario. Two objects were placed on a shelf and, using the robot, the users were tasked to retrieve the objects to a basket on a nearby table, which was close to the stored *Home* position. Each user could select the order in which to retrieve the objects, so the system could not be adjusted to the specific setting. After each successful retrieval, the robot was reset to a starting pose to make the trials comparable.

We developed a user interface (UI) for our adaptive DoF control on the same principles as the munevo DRIVE[14] system for controlling an electric wheelchair with head gestures through a Google Glass[6]. This includes mode selection by flicking or nodding motions of the head, and control inputs by tilting the head. The UI can be seen in the top right of Figure 1. The DoF currently controlled is shown in the center, with the new suggestion being highlighted on the side. The cardinal DoFs can be reached by nodding. DoFs are represented with a set of simplified 3D arrows. The use of this interface aimed to provide a sense of a realistic interaction experience for our study participants.

During the study, each participant compared our control to the use of only the cardinal control, both using the same interface. The order of controls was switched for each user. In addition to extensive explanations, the users were also given time to get used to both control methods, and a test task (grasp a held bottle) for introduction. For the latter, they were verbally guided and assisted by the study administrator.

5.1 Results

Our user group of 18 people was aged between 20 and 34 years (25.8 ± 4.2). Of these, 8 reported their gender as female, 8 as male, and none as non-binary, with two choosing not to reply. All test subjects were able-bodied and reported no personal context to the field of care. Most of them (15) regularly used joysticks or keyboards, with 6 users working with robots on a weekly basis.

For the evaluation, we examined the time between the first motion of the robot and the start of opening the fingers to release the object at the target position. Figure 5 shows the execution times over all users with either control method, separated by execution order. For the first object (i.e. when there is very little experience with the control), it can be seen that they are slower with the adaptive control. Even though only slightly better than the classic approach, there is a noticeable improvement of the adaptive duration for the second object, when compared to the first. This is independent of the order the controls were presented in, although the effect is more prominent if classic was used first.

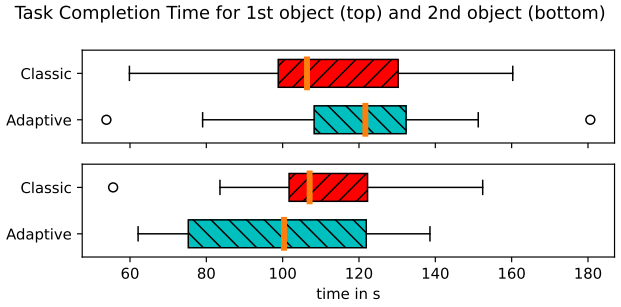

Figure 5: Task Completion Times of all users, separated by object

Another factor for the usability of an assistive control concept is the necessity of mode switches. Depending on the interface and the physical abilities of the user, mode switches can be difficult, exhausting, or time-consuming. It is therefore an important metric of the system. Figure 6 shows the number of mode switches for each control, again separated by execution order. This clearly shows a significant reduction in mode switches for the adaptive control for both objects.

A slight variation of this can be seen in Figure 7, which shows the number of user-interface selections necessary to reach the actual modes to control in. This differs, as the limited input options of the user interface requires the user to skim through various options to reach the desired control mode. The difference between classic and adaptive is even greater, showing reduced necessary skimming

Authors´ accepted version. https://doi.org/10.1145/3453892.3453895

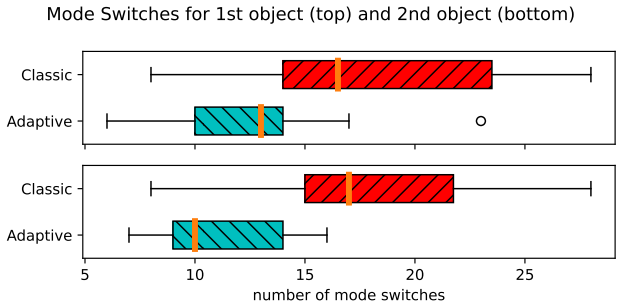

Figure 6: Number of Mode Switches of all users, separated by object

with the adaptive control. A clear training effect can be observed by the reduction of variability (broader range of interquartiles) from the first object to the second.

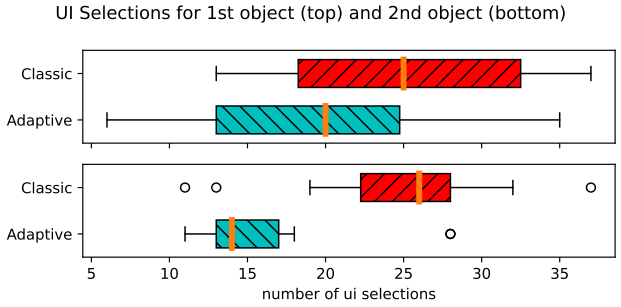

Figure 7: Number of UI Selection Gestures of all users, separated by object

The qualitative user responses had a clear preference: 14 chose the adaptive control and only 4 the classic. The users gave different reasons for this: Some users reported the adaptive control to be simpler, even if they require more familiarization, whereas others simply felt the visual user interface more compact (less mode options), therefore allowing for a better overview. The latter also caused for fewer necessary mode switches, which were sometimes physically exhausting for users. Multiple users explicitly praised the diagonal options of the adaptive control, assisting to move direction to a target, which in turn was an issue for a smaller subset, which had difficulty subconsciously grasping the directions of the more complex arrows.

For this user group, the classic side had a clear advantage, as one did not need to understand the arrows, but could instead learn the modes and positions by heart for this control. In addition, one user reported that the directions suggested by the adaptive control interface often did not fit his wishes.

The conducted NASA-TLX questionnaire was not conclusive: All categories showed very similarly distributed user responses. Based on the interviews, this can be traced back to differing perspectives. For example, for some users the mental demand of the adaptive control was higher because the arrows changed and one had to adapt, whereas others found this to be a reduction of mental load, as the more direct options removed intermediate steps and allowed for more streamlined executions. The physical demand was mostly identical, with some users reporting less strain on the adaptive control because of the reduction in mode changes. The results of the NASA-TLX can be seen in Figure 8.

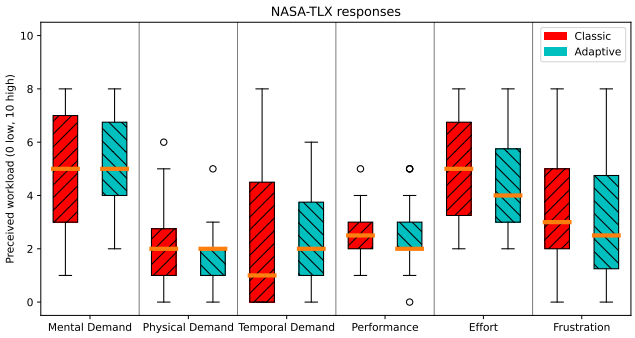

Figure 8: NASA-TLX User Responses

6 COMPARISON TO A HYPOTHETICAL END-TO-END LEARNING APPROACH

It is tempting to learn the desired $p(U_t|Z_{1:t})$ end-to-end from data. Indeed, in previous work we recorded the DORMADL dataset[4] exactly for this purpose. However, investigation of the data revealed considerable complications for such an approach. These motivated our investigation into a more classical engineering alternative reported in this paper and shall be discussed here.

First, as the camera is mounted slightly behind the hand (Fig. 1), it often sees only a small part of the scene. However, the motion is often motivated by something outside the view, e.g. when transitioning from one object to another on a table. This requires the system to memorize object positions to effectively predict motion. In our system this is done analytically with depth data and forward kinematics. An end-to-end learned system would have to learn this connection, which is not easy and of little value, since it is well described analytically. It would also require operating on image sequences not single images.

Second, we observed that users exhibit a large degree of arbitrariness in how they perform a certain motion. This makes it hard to predict. It is actually not necessary for the application to predict precisely which motion variant is desired as any useful variant is fine. However, this is not captured by typical least-squares or maximum likelihood losses.

Third, we observed that the presumably well predictable motions are reaching motions to grasp an object. As we showed, these are low-hanging fruits that do not need deep learning but are well realized geometrically.

Forth, an advantage of deep learning is that it would implicitly learn to understand the environment from images, which cannot be done well analytically. However, as shown in our system, this part can be covered by a network with a more specialized role (segment anything) that has actually been trained on much more data.

Lastly, as most often, the analytical approach is more transparent, because it is build of parts that have a human defined interface. It is also easier adaptable, because the heuristic definition of behaviors can be changed, while changing a learned one requires collecting a new dataset reflecting the desired changes in behavior.

Overall, while we don't want to rule out an end-to-end learned approach, we claim that the proposed method is well suited for the considered task.

7 CONCLUSION

We have presented a system that suggests degrees of freedom to the user of an assistive robot arm. It models the distribution of what motion the user probably wants to do as a mixture of heuristically defined behaviors. Some of these, in particular the approaching and grasping behavior, incorporate sensor data, mostly in a geometric way. This has been shown as a viable alternative to an end-to-end learning idea.

The technical user study showed the viability of the system in a realistic scenario with a promising perspective on mode switch reduction. Also, the qualitative feedback of the users displayed clear preferences for the new control system, especially during the direct approach of objects.

Future work is to investigate, whether more general behaviors can be implemented this way and whether they are actually predictable to an extent that allows suggesting a DoF to the user. Also, the study needs to be repeated with the targeted user group.

ACKNOWLEDGMENTS

This work was supported by the project *DOF-Adaptiv*, funded by the German Federal Ministry of Education and Research BMBF (Bundesministerium für Bildung und Forschung), FKZ 16SV8563. It also received support from the *REXASI-PRO* H-EU project, call HORIZON-CL4-2021-HUMAN-01-01, under the Grant agreement ID 101070028.

REFERENCES

[1] Brenna D Argall. 2013. Machine learning for shared control with assistive machines. In *Proceedings of ICRA Workshop on Autonomous Learning: From Machine Learning to Learning in Real world Autonomous Systems, Karlsruhe, Germany.*

[2] Rodney A. Brooks. 1991. Intelligence without representation. *Artificial Intelligence* 47, 1 (1991), 139–159. https://doi.org/10.1016/0004-3702(91)90053-M

[3] Hongwu Wang Cheng-Shiu Chung and Rory A. Cooper. 2013. Functional assessment and performance evaluation for assistive robotic manipulators: Literature review. *The Journal of Spinal Cord Medicine* 36, 4 (2013), 273–289. https://doi.org/10.1179/2045772313Y.0000000132

[4] Felix Goldau, Yashaswini Shivashankar, Annalies Baumeister, Lennart Drescher, Patrizia Tolle, and Udo Frese. 2023. DORMADL - Dataset of Human-Operated Robot Arm Motion in Activities of Daily Living. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE Press, Piscataway, NJ, 11396–11403. https://doi.org/10.1109/IROS55552.2023.10341459

[5] Felix Ferdinand Goldau and Udo Frese. 2021. Learning to Map Degrees of Freedom for Assistive User Control: Towards an Adaptive DoF-Mapping Control for Assistive Robots. In *Proceedings of the 14th PErvasive Technologies Related to Assistive Environments Conference* (Corfu, Greece) *(PETRA '21)*. Association for Computing Machinery, New York, NY, USA, 132–139. https://doi.org/10.1145/3453892.3453895

[6] Google. 2019. Glass Enterprise Edition 2.

[7] Laura V. Herlant, Rachel M. Holladay, and Siddhartha S. Srinivasa. 2016. Assistive teleoperation of robot arms via automatic time-optimal mode switching. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE Press, Piscataway, NJ, 35–42. https://doi.org/10.1109/HRI.2016.7451731

[8] Intel Corporation. 2022. Intel Realsense - Product Family D400 Series. (last visited on 13.02.2023). https://www.intelrealsense.com/wp-content/uploads/2022/11/Intel-RealSense-D400-Series-Datasheet-November-2022.pdf

[9] Kinova. 2021. Jaco assistive robot - User guide. (last visited on 13.02.2023). https://assistive.kinovarobotics.com/uploads/EN-UG-007-Jaco-user-guide-R05.pdf "EN-UG-007-r05-202111".

[10] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. 2023. Segment Anything. arXiv:2304.02643 [cs.CV]

[11] Dylan P. Losey, Krishnan Srinivasan, Ajay Mandlekar, Animesh Garg, and Dorsa Sadigh. 2020. Controlling Assistive Robots with Learned Latent Actions. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE Press, Piscataway, NJ, 378–384. https://doi.org/10.1109/ICRA40945.2020.9197197

[12] Veronique Maheu, Philippe S. Archambault, Julie Frappier, and François Routhier. 2011. Evaluation of the JACO robotic arm: Clinico-economic study for powered wheelchair users with upper-extremity disabilities. In *2011 IEEE International Conference on Rehabilitation Robotics*. IEEE Press, Piscataway, NJ, 1–5. https://doi.org/10.1109/ICORR.2011.5975397

[13] Abhilash Alexander Miranda, Yann-Aël Le Borgne, and Gianluca Bontempi. 2008. New routes from minimal approximation error to principal components. *Neural Processing Letters* 27 (2008), 197–207. https://doi.org/10.1007/s11063-007-9069-2

[14] Horst Penkert, Julius C. Baron, Konstantin Madaus, Wolfgang Huber, and Achim Berthele. 2021. Assessment of a novel, smartglass-based control device for electrically powered wheelchairs. *Disability and Rehabilitation: Assistive Technology* 16, 2 (2021), 172–176. https://doi.org/10.1080/17483107.2019.1646817

[15] Rudolph Van Der Merwe and Eric A. Wan. 2004. *Sigma-point kalman filters for probabilistic inference in dynamic state-space models*. Ph. D. Dissertation. Oregon Health & Science University. AAI3129163.

[16] Jörn Vogel, Katharina Hertkorn, Rohit U. Menon, and Máximo A. Roa. 2016. Flexible, semi-autonomous grasping for assistive robotics. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE Press, Piscataway, NJ, 4872–4879. https://doi.org/10.1109/ICRA.2016.7487692

[17] W. Grey Walter. 1950. An Imitation Of Life. *Scientific American* 182, 5 (1950), 42–45. http://www.jstor.org/stable/24967456

[18] Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. 2023. Fast Segment Anything. arXiv:2306.12156 [cs.CV]