



# Dual filtering in operational and joint spaces for reaching and grasping

Léo Pio-Lopez, Jean-Charles Quinton, Youcef Mezouar

## ► To cite this version:

Léo Pio-Lopez, Jean-Charles Quinton, Youcef Mezouar. Dual filtering in operational and joint spaces for reaching and grasping. *Cognitive Processing*, 2015, 16 (S1), pp.293-297. 10.1007/s10339-015-0710-0 . hal-01966562

**HAL Id: hal-01966562**

**<https://hal.science/hal-01966562>**

Submitted on 28 Dec 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Dual filtering in operational and joint spaces for reaching and grasping

Léo Pio-Lopez · Jean-Charles Quinton · Youcef Mezouar

Received: date / Accepted: date

**Abstract** To study human movement generation, as well as to develop efficient control algorithms for humanoid or dexterous manipulation robots, overcoming the limits and drawbacks of inverse kinematics based methods is needed. Adequate methods must deal with high dimensionality, uncertainty, and must perform in real-time (constraints shared by robots and humans). This paper introduces a Bayesian filtering method, hierarchically applied in the operational and joint spaces to break down the complexity of the problem. The method is validated in simulation on a robotic arm in a cluttered environment, with up to 51 degrees of freedom.

**Keywords** Bayesian filtering · reaching · operational space · joint space · grasping

## 1 Introduction

A clear trend to include robots in our everyday environments is emerging. These robots need to find their way and interact with objects and humans in uncontrolled, uncertain and often cluttered environments, way different from the structured environment of a manufacturing plant. In manipulation robotics, using the arm to reach a target in the operational space or to place fingers on objects is fundamental. This task generally requires to specify a desired end-effector pose (position and orientation) and to determine the joint parameters

allowing the robot to reach it, a problem also known as inverse kinematics (IK). Several classical approaches exist to solve this problem, which rely on algebraic (Fu, Gonzalez, & Lee, 1987), iterative (Korein & Badler, 1982) and geometric methods (Lee, 1982).

Nevertheless for healthy human adults, reaching a target feels like an easy task, yet involves many degrees of freedom (DoFs) and the online generation of complex trajectories in the joint space. To fully understand human movement generation, as well as to develop efficient algorithms for humanoid or dexterous manipulation robots (that share the same constraints), overcoming the limits and drawbacks of classical control approaches is required.

Without resorting to inverse kinematics (that may be excessively time consuming or lead to rough approximations), we want to show that it is possible to control high dimensional systems by simulating and predicting the outcome of local actions (forward model only), as long as the problem complexity is broken down into smaller subspaces. We here focus on the decomposition of the global sensorimotor problem of reaching one or several targets with one or several effectors in the operational and joint spaces (or visual and motor spaces in cognitive science). We rely on a probabilistic method for generating reaching movements in complex settings. Specifically, we apply (approximate) Bayesian filtering successively in the operational and joint spaces. Operational filtering permits to define an initial rough trajectory in the operational space (avoiding obstacles), which is then refined by joint filtering, allowing direct control in the joint space while enforcing joint limits. Additionally, this method is biologically plausible, as such filtering has been proposed to be implemented in the brain (Deneve, Duhamel, & Pouget, 2007).

The method was validated in simulation on a set of scenarios. With strong spatial constraints (e.g. obstacles), the method succeeds in finding a trajectory where inverse kine-

---

Léo Pio-Lopez  
Clermont University, Pascal Institute, Clermont-Ferrand, France  
E-mail: leo.pio.lopez@gmail.com

Jean-Charles Quinton  
Clermont University, Pascal Institute, Clermont-Ferrand, France  
CNRS (UMR 6602), Pascal Institute, Aubiere, France

Youcef Mezouar  
IFMA, Pascal Institute, Aubiere, France  
CNRS (UMR 6602), Pascal Institute, Aubiere, France

matics methods fail. Relying on the operational space thus seems promising to reduce the complexity of movement generation in the joint space.

## 2 Model

Several problems in science and engineering involve estimating some (hidden) states given observables or noisy measurements. This problem can be solved using Bayesian filtering methods (Doucet, De Freitas, & Gordon, 2001), which we adopt in this paper and apply to robotics. Two classes of variables are thus involved :  $S_0, \dots, S_t$  which are the state variables considered on a time horizon ranging from 0 to  $t$ . And  $O_0, \dots, O_t$  the time series of observation variables on the same horizon. The decomposition of the joint probability of this class of model is:

$$P(S_0, \dots, S_t, O_0, \dots, O_t) = P(S_0)P(O_0 | S_0) \prod_{i=1}^t P(S_i | S_{i-1})P(O_i | S_i) \quad (1)$$

which can be understood by considering the three following righthand terms:

- $P(S_0)$  is a prior on the state at time  $t = 0$ .
- $P(S_t | S_{t-1})$  is the "evolution model", which corresponds to the knowledge concerning the possible transitions between two time steps.
- $P(O_t | S_t)$  is the "observation model", which corresponds to the information we can observe and thus exploit if the state is  $S_t$ .

The question to answer with Bayesian filtering is how to determine the probability distribution over the states at time  $t$  knowing the sequence of observations, i.e.  $P(S_t | O_0, \dots, O_t)$ . We can decompose this probability distribution in a recursive way (Doucet et al., 2001), using the predict (Eq.2) and update steps (Eq.3).

$$P(S_t | O_{0:t-1}) = \int P(S_t | S_{t-1})P(S_{t-1} | O_{0:t-1})dS_{t-1} \quad (2)$$

$$P(S_t | O_{0:t}) = \frac{P(O_t | S_{0:t} O_{0:t-1})P(S_{0:t-1} | O_{0:t-1})}{\int P(O_t | S_t)P(S_t | O_{0:t-1}), dS_t} \quad (3)$$

These probability density functions are hard to compute as they imply the calculation of high-dimensional and complex integrals. We need approximation in order to use Bayesian filtering in real-time. The approximation we have chosen is particle filtering which is an instance of Sequential Monte Carlo (SMC) methods (Doucet et al., 2001).

Since this is not sufficient to make this method directly applicable for online movement planning on systems with

a large number of DoFs, we apply Bayesian filtering in a two steps procedure, successively in the operational space and joint space. The first step is thus about generating a rough 3D trajectory, similar to what humans can imagine and internally simulate when they want to catch an object. The resulting trajectory is then used as a guiding thread to follow during step 2, where Bayesian filtering is applied in the joint space. This 3D trajectory permits to define a set of sub-targets to reach for the arm before attaining the final target. The following subsections describe the evolution models and observation models used in the operational and joint spaces.

### 2.1 Evolution model in the operational space

In the operational space, the state variables are the position (in 3D world coordinates) and the observation variables correspond to a set of constraints. In the examples provided in the results, the position will correspond to the center of the hand. The evolution prior describes the nature of the movement, and we chose here a basic random walk (see Eq.4) instead of a more complex model based prior. Nevertheless, the prior could be learned or improved, for instance relying on human demonstration (Argall, Chernova, Veloso, & Browning, 2009). In any case, we want to simulate a trajectory between the actual position of the end-effector of the robot and a target in the operational space while coarsely avoiding obstacles.

$$P(S_t | S_{t-1}) = \text{random walk} = \Phi(S_t S_{t-1}, \sum_S) \quad (4)$$

A gaussian noise model controls the actual displacement in operational space. We deliberately used a Gaussian noise model in order to find trajectories in complex environments in an online manner even if we have to sacrifice the optimality of the trajectory. The covariance matrix does not enforce the limits of the joints of the robot.

#### 2.1.1 Evolution model in the joint space

In the joint configuration space and for the proof of concept purpose of the present paper, the state variables again correspond to the joint configuration of the center of the hand. The evolution model has also been chosen in order to find a trajectory avoiding objects in a coarse manner. The equation is therefore the same as in the operational space (see Eq.4).

A gaussian noise model also control the displacements for the simulation of the trajectory. However we used this time a fine tuning method for this evolution model. The covariance matrix  $\sum_S$  enforces the limits of each joint. The sampling of particles is performed following this model and

if the kinematic properties of the joints are not enforced for any of the  $S_i$  due to random walk, we assigned the variables of interest of the corresponding joint configuration to its lower or upper limit depending of the limit that has been reached. This approach avoids the case where the algorithm enters in an endless rejection of values for the joint configuration, because the generation of particles is confined to a physically unreachable part of the space. Once again, as it corresponds to a low level body schema representation of the system (Sturm, Plagemann, & Burgard, 2009), this model could be learned (e.g. via human demonstration or motor babbling), aiming for more naturalistic movements and to limit the useless rejection of particles.

### 2.1.2 Observation model in the operational space

The observation variables in this case correspond to several constraints the robot has to enforce. The first constraint is to avoid obstacles to guarantee its own integrity and the second one is to generate a movement as efficient as possible towards the target. In this paper, efficiency is simply defined as the minimization of the distance required to approach the target. In a probabilistic way, this can be written as:

$$P(O_t | S_t) \propto \begin{cases} \exp(-|X_{endeffector} - X_{target}|) \\ 0 \text{ if obstacles} \end{cases} \quad (5)$$

This formulation of the observation model is really flexible as we can add as many constraints as we want. We can also associate weights to the constraints in order to implement a prioritization of the constraints.

### 2.1.3 Observation model in the joint space

The constraints are the same as for the simulated trajectory in the operational space, but in order to demonstrate how the hand and fingers should be preferably used only when the target/object has been approached, we weight the constraints as follows:

$$P(O_t | S_t) \propto \begin{cases} \exp(-2 * |D_1|) \cdot \prod_{i=2}^n \exp(-|D_i|) \\ 0 \text{ if obstacles} \end{cases} \quad (6)$$

where  $D_1$  is the distance from the hand to the target and  $D_i$  with  $i = 2, \dots, n$  are the distances from the fingertips to the target. The exponential function allows us to define a probability density of the distances from the targets. The weighted combination of these constraints allows the algorithm to focus first on the reaching of the target object and in a second time to grasp it, by setting the parameter. As many targets as we want can be added with this algorithm, by simply adapting the considered end effectors and associated distances. Although any heuristics could be used, a

direct model of the robot allows the correct computation of the distances from the arm and fingers to the target. We have to precise that for the following tasks the targets to reach are the sub-targets defined after the filtering in the operational space. A sub-target is considered reached when the distance between the end effector and the sub-target decreases below some error threshold, as the filtering in the operational space does not enforce the joint limits. The sub-target the closest from the final target is then selected until the arm reaches the final one.

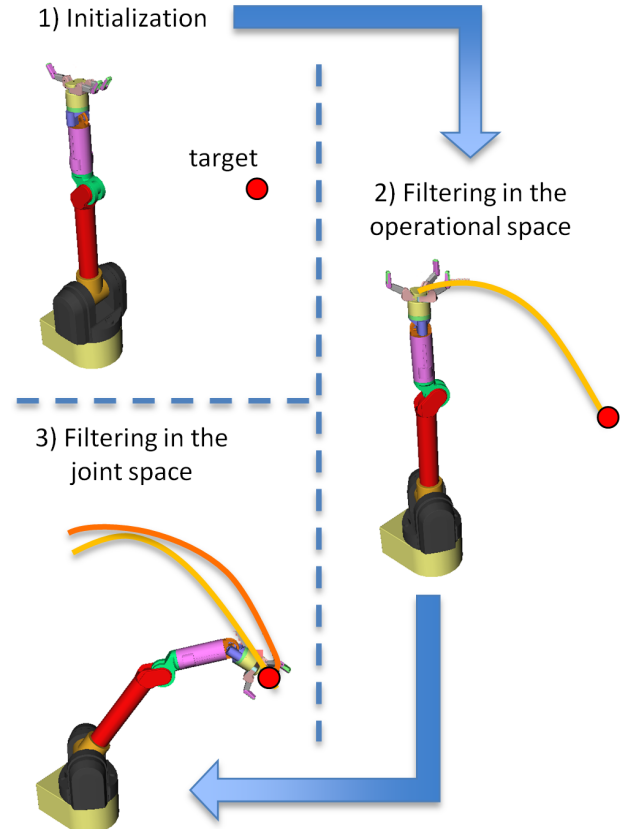


Fig. 1: Description of the method. The first step is the initialization of all the variables. The second step consists in the filtering in the operational space. This permit to define a first trajectory to follow for control like a guiding thread. The last step is the the filtering in the joint space which is used for controlling the robotic system.

## 3 Results

We applied our algorithm on several examples in order to show the advantages of dividing the problem into the operational and joint spaces. All the simulations have been developed with OpenRave (Diankov & Kuffner, 2008). We start by demonstrating in the first example why the coupled Bayesian filtering is more efficient than Bayesian filtering in the joint space alone. We then turn to a second example

demonstrating the scalability of the algorithm, by applying it on a 51 DoFs robotic systems. In both cases, our method allows the online generation of reaching movements.

### 3.1 Cluttered environment

We demonstrate here that the hierarchical decomposition of the problem and thus the application of dual Bayesian filtering is clearly helpful to find solutions in cluttered environments. We defined an hard to reach target, located under a large obstacle in a simulated human environment, while the manipulator has to simultaneously deal with several other obstacles and its own joint limits in order to reach it.

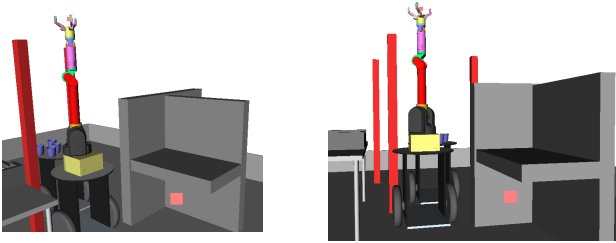


Fig. 2: Left: initial pose of the arm and target. Right: Other orientation of the scene. The target for reaching is in orange and is located under the bench.

On 50 trials, applying the algorithm directly and only in the joint space leads to 0% of reaching successes. The manipulator always converges to a local extrema above the main obstacle and cannot escape it, due to a lack of global planification.

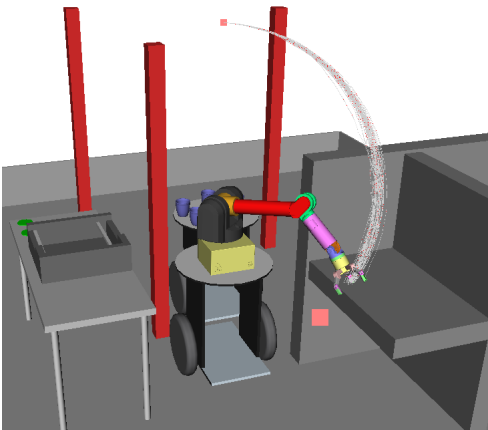


Fig. 3: 50 trajectories with Bayesian filtering applied into the joint space only. The arm always hit the bench.

Although the same reason explains why the application of dual Bayesian filtering only reaches a success rate of 30%,

the definition of the guiding thread in operational space significantly improves performance, even when it cannot be followed closely.

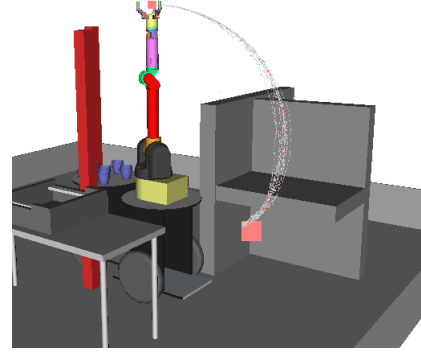


Fig. 4: 50 trajectories with Bayesian filtering applied into the operational space first and then into the joint space.

### 3.2 Redundant arm

On an ultra-redundant system, standard inverse kinematics methods can have problem to compute a solution in real-time. Our method can be applied on a such robot with a high number of DoFs without a significant increase in computational cost. Indeed, the computational complexity of our algorithm only depends on the number of particles (Doucet et al., 2001), but would nevertheless requires an exponential increase in the number of particles if filtering was applied in the joint space only, because of the curse of dimensionality. Nevertheless, filtering in the operational space (always 3D) again acts as a scaffolding step, able to reduce the complexity of the search in the high dimensional space. Using the previous setup, except for a custom WAM arm with 51 DoFs, we obtained 64% of reaching success.

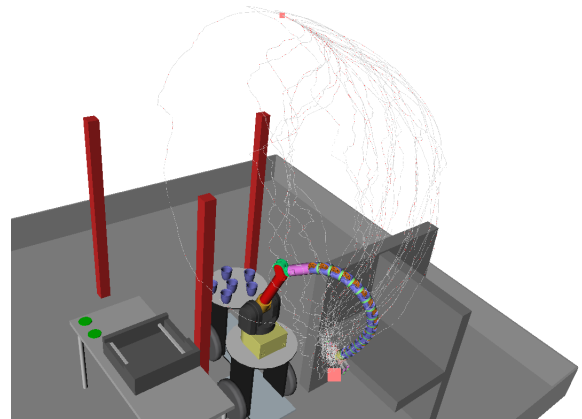


Fig. 5: Trajectories with the 51 DoFs arm.

#### 4 Conclusion and perspectives

We presented in this paper a biologically plausible Bayesian filtering method for reaching and grasping. We applied it on two proof-of-concept simulated setups to show its advantages as an alternative to inverse kinematics. Our results demonstrate that under some conditions, the problem of reaching can be simplified by breaking it down into two spaces, applying Bayesian filtering successively in the operational and then joint space. This hierarchical decomposition generates a reference trajectory in a low dimensional space, later exploited to constrain the exploration of the high dimensional space. This method does not require an inverse model, a high number of DoFs can be controlled, and can be easily extended to deal with a large number of targets and effectors. Additionally, the method performs in real-time, and computation time could be easily reduced through parallelization (natural for a particle filter approximation).

Several extensions are possible, including its applications on more complex robots with several kinematic chains. Also, basic distance to target functions are used for the observation models, but other functions could be used in conjunction with the distance in order for instance to minimize energy consumption. An arbitrary random walk is finally used for the evolution model, but these models could be learned from, improving the sampling of particles, allowing the generation of more human-like movements and further testing the biological plausibility of such method.

#### References

- Argall, B. D., Chernova, S., Veloso, M., & Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5), 469–483.
- Deneve, S., Duhamel, J.-R., & Pouget, A. (2007). Optimal sensorimotor integration in recurrent cortical networks: a neural implementation of kalman filters. *The Journal of neuroscience*, 27(21), 5744–5756.
- Diankov, R., & Kuffner, J. (2008). Openrave: A planning architecture for autonomous robotics. *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-08-34*, 79.
- Doucet, A., De Freitas, N., & Gordon, N. (2001). An introduction to sequential monte carlo methods. In *Sequential monte carlo methods in practice* (pp. 3–14). Springer.
- Fu, K. S., Gonzalez, R., & Lee, C. G. (1987). *Robotics: Control sensing vision and intelligence*. McGraw-Hill.
- Korein, J. U., & Badler, N. I. (1982). Techniques for generating the goal-directed motion of articulated structures. *IEEE Computer Graphics and Applications*, 2(9), 71–81.
- Lee, C. G. (1982). Robot arm kinematics, dynamics, and control. *Computer*, 15(12), 62–80.
- Sturm, J., Plagemann, C., & Burgard, W. (2009). Body schema learning for robotic manipulators from visual self-perception. *Journal of Physiology-Paris*, 103(3), 220–231.