Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Learning to Catch Moving Objects with Reduced Impulse Exchange [1]

A. S. Phung, J. Malzahn, F. Hoffmann and T. Bertram

Institute of Control Theory and System Engineering,
Technische Universität Dortmund, 44221 Dortmund, Germany

Abstract: The paper presents a learning from demonstration approach to the catching of moving objects with a robot manipulator. The work explicitly reduces the impulse exchange by minimizing the relative velocity during the contact phase by learning relative relation between the object and the catcher instead of learning separate forward model of the object and trajectory generator for the robot. This contributes to the damage prevention on both, the object and the robot. The demonstrated catching movements are modelled by a Gaussian mixture model (GMM), which describes the probability distribution over the demonstrated data set. Gaussian mixture regression (GMR) is employed for motion prediction. A timing controller is designed to trade-off the catch location and the catching time. The learning scheme comprises the relative position and velocity profile between the object and the catcher. The approach allows to generate the movement of the catcher with guaranteed position and velocity convergence to the reference inferred from the position and velocity of the object. Experimental results with a five degree of freedom arm and a Photonic-Mixing-Device (PMD) camera for object detection validate the approach.

1. INTRODUCTION

Catching moving objects by a robot can be used in many applications. Catching work pieces moving on a conveyer belt speeds up cycle times and may reduce fatigue in the belt mechancis through avoiding repeated belt decelleration, waiting and acceleration phases. Another example is the cooperation between manipulators and mobile robots in intralogistics. Mobile robots transport goods while the manipulators take and organize them on shelves without requiring the mobile robots to stop. Conversely the manipulators can place the goods on moving mobile robots.

A crucial skill is the minimization of potentially harmful impact forces during the contact phase of the catching task. According to Kajikawa et al. [1999], humans generally accomplish this quite well, even with previously unknown moving objects. This motivates the learning from demonstration approach proposed in this paper. A manipulator realizes the catching task for a moving object and mimics human demonstrations to minimize the potential harm to the object.

Robot "learning from demonstration" or "robot programming by demonstration" describe a technique to enable quick and intuitive robot programming even for non-robotic specialists. The technique experiences great attention in motion planning for robots (Schaal et al. [2007], Aleotti and Caselli [2006], Hoffmann et al. [2008], Phung et al. [2011]). Aleotti and Caselli [2006] cluster the demonstrated trajectories before they use hidden markov models (HMM) and rational B-splines to select and approximate a desired trajectory. Schaal et al. [2007], Hoffmann et al. [2008] and Phung et al. [2011] use the dynamical movement primitives (DMP) to model the demonstrated trajectory. The model allows to estimate the subsequent states. The advantages of DMP are the guaranteed convergence to a goal point and the straightforward adaptivity in the spatial as well as the temporal domain by modification of the goal point and the time constant of the dynamical system. A similar approach to learn, represent and generalize point-to-point movements is based on autonomous dynamical systems (DS) and proposed by Calinon et al. [2007], Khansari-Zadeh et al. [2010], Gribovskaya et al. [2011]. In contrast to conventional robot motion planning methods using polynomial functions of time or DMP systems, DS do not depend on an explicit time variable. Hence, there is no need for temporal adaptation, which renders the mapped dynamics insensitive to temporal perturbations of the actual task.

For the measurement of the object positions as well as trajectory prediction Frese et al. [2001], Riley and Atkeson [2002], Namiki and Ishikawa [2003] employ vision systems. Analogously, this work uses a Photonic-Mixing-Device (PMD) camera, which also provides scene depth measurements. Namiki and Ishikawa [2003] directly map the object trajectory to the joint angle trajectory. Frese et al. [2001], Riley and Atkeson [2002] predict the catching point and perform a point-to-point movement to intercept the object at this location. Malzahn et al. [2012] implement a similar concept on a multi-elastic-link robot arm to catch multiple balls subsequently thrown by a human.

Kim et al. [2010] also apply learning from demonstration to catch a flying ball. DS are used to encode and generate both, the object as well as the robot trajectory. The ball trajectory is predicted by forward integration of the DS learned offline from several throws. A catch point is determined by checking the intersection line of the estimated ball trajectory with the robot workspace. The

[1] This work is supported by Karl Kolle Foundation.

intercept point represents the attractor of the DS learned for the robot motion from human demonstrations. Since the DS is a time-independent system, the authors use a speed-up or slow-down factor for the DS so that the robot is able to catch the ball in time.

In this paper a Gaussian Mixture Model is combined with a nonlinear autonomous dynamical system to model the motion task from demonstrations. The idea is to calculate the probability distribution between the motion variables, so that the generalization to new trajectories is feasible through Gaussian Mixture Regression (GMR). A variety of GMR techniques exist: Expectation-Maximization (EM) (Dempster et al. [1977]), Binary Merging (BM) (Khansari-Zadeh and Billard [2010]). This contribution applies Stable Estimator of Dynamical Systems (SEDS) (Khansari-Zadeh et al. [2010]) for the identification of the GMM parameters from the demonstrations. The work extends the approach presented in Kim et al. [2010] by considering not only the relative position between the robot and the object, but also their relative velocity right before the contact. Instead of focousing at the predicted poses of the moving object, we directly use the actual velocity of the object to generate the robot motion. This increases the robustness of the system with respect to unexpected object movements, which otherwise would require the computationally intensive reestimation of the complete object trajectory.

The next section presents the GMM formulation and the reproduction as well as generation of the movement using GMR. Section 3 shows the simulation results for catching the moving object in two cases: with explicit minimization of the impulse exchange during contact in subsection 3.1 and with implicit time dependency in subsection 3.2. The experimental results are illustrated in section 4. Conclusions are drawn in section 5.

2. MOTION GENERALIZATION WITH GAUSSIAN MIXTURE MODEL

This section presents the autonomous dynamical system $\dot{\mathbf{x}} = f(\mathbf{x}, \boldsymbol{\theta})$ based on gaussian mixture model (GMM), where $\mathbf{x}$ is the state variables and $\boldsymbol{\theta}$ the parameters of the system. A GMM comprising K Gaussians is defined by the probability distribution function:

$$p\left(\boldsymbol{\xi}_i, \dot{\boldsymbol{\xi}}_i\right) = \sum_{k=1}^{K} p(k)\, p\left(\boldsymbol{\xi}_i, \dot{\boldsymbol{\xi}}_i | k\right), \tag{1}$$

over a set of N demonstrated trajectories $\left\{\boldsymbol{\xi}_i^{\mathrm{T}}, \dot{\boldsymbol{\xi}}_i^{\mathrm{T}}\right\}$, with $i = 1..N$, and

$$\begin{aligned} p(k) &= \pi^k, \\ p\left(\boldsymbol{\xi}_i, \dot{\boldsymbol{\xi}}_i | k\right) &= G\left(\boldsymbol{\xi}_i, \dot{\boldsymbol{\xi}}_i; \boldsymbol{\mu}^k, \boldsymbol{\Sigma}^k\right) \\ &= \frac{1}{\sqrt{(2\pi)^{2D}\left|\boldsymbol{\Sigma}^k\right|}} e^{-\frac{1}{2}\left((\mathbf{d}_i - \boldsymbol{\mu}^k)^T (\boldsymbol{\Sigma}^k)^{-1} (\mathbf{d}_i - \boldsymbol{\mu}^k)\right)}. \end{aligned}$$

Therein D is the dimension of $\boldsymbol{\xi}_i$, $\mathbf{d}_i = \left[\boldsymbol{\xi}_i, \dot{\boldsymbol{\xi}}_i\right]^{\mathrm{T}}$ is the dataset of the demonstrations and π^k are the prior probabilities. The Gaussian G is defined by the mean vector $\boldsymbol{\mu}^k$ and covariance matrix $\boldsymbol{\Sigma}^k$:

$$\boldsymbol{\mu}^k = \begin{bmatrix} \boldsymbol{\mu}_{\boldsymbol{\xi}}^k \\ \boldsymbol{\mu}_{\dot{\boldsymbol{\xi}}}^k \end{bmatrix} \qquad \boldsymbol{\Sigma}^k = \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{\xi}}^k & \boldsymbol{\Sigma}_{\boldsymbol{\xi}\dot{\boldsymbol{\xi}}}^k \\ \boldsymbol{\Sigma}_{\dot{\boldsymbol{\xi}}\boldsymbol{\xi}}^k & \boldsymbol{\Sigma}_{\dot{\boldsymbol{\xi}}}^k \end{bmatrix}. \tag{2}$$

.

Assume that the demonstrated movements are modelled by a GMM expressed by (1). Gaussian mixture regression (GMR) infers the expected distribution from a given input and a given Gaussian distribution. This way, it is feasible to reproduce a previously seen movement or to generate new movements. In conjunction with a dynamical system, the GMR is applied to compute the probability of $\dot{\boldsymbol{\xi}}_i$ from the actual state $\boldsymbol{\xi}_i$ by using $p\left(\dot{\boldsymbol{\xi}}_i \mid \boldsymbol{\xi}_i\right)$ estimated from the GMM as follows:

$$p\left(\dot{\boldsymbol{\xi}}_i \middle| \boldsymbol{\xi}_i\right) = \sum_{k=1}^{K} h^k p\left(\dot{\boldsymbol{\xi}}_i | \boldsymbol{\xi}_i, k\right) = \sum_{k=1}^{K} h^k G\left(\hat{\boldsymbol{\mu}}^k, \hat{\boldsymbol{\Sigma}}^k\right), \tag{3}$$

where

$$\begin{aligned} h^k &= p\left(k | \boldsymbol{\xi}_i, \dot{\boldsymbol{\xi}}_i\right) \\ &= \frac{p(k)\, p\left(\boldsymbol{\xi}_i | k\right)}{\sum_{j=1}^{K} p(j)\, p\left(\boldsymbol{\xi}_i | j\right)} = \frac{\pi^k G\left(\boldsymbol{\xi}_i; \boldsymbol{\mu}_{\boldsymbol{\xi}_i}^k, \boldsymbol{\Sigma}_{\boldsymbol{\xi}_i}^k\right)}{\sum_{j=1}^{K} \pi^j G\left(\boldsymbol{\xi}_i; \boldsymbol{\mu}_{\boldsymbol{\xi}_i}^j, \boldsymbol{\Sigma}_{\boldsymbol{\xi}_i}^j\right)}, \end{aligned} \tag{4}$$

$$\hat{\boldsymbol{\mu}}^k = \boldsymbol{\mu}_{\dot{\boldsymbol{\xi}}}^k + \boldsymbol{\Sigma}_{\dot{\boldsymbol{\xi}}\boldsymbol{\xi}}^k \left(\boldsymbol{\Sigma}_{\boldsymbol{\xi}}^k\right)^{-1} \left(\boldsymbol{\xi}_i - \boldsymbol{\mu}_{\boldsymbol{\xi}}\right), \tag{5}$$

$$\hat{\boldsymbol{\Sigma}}^k = \boldsymbol{\Sigma}_{\dot{\boldsymbol{\xi}}}^k - \boldsymbol{\Sigma}_{\dot{\boldsymbol{\xi}}\boldsymbol{\xi}}^k \left(\boldsymbol{\Sigma}_{\boldsymbol{\xi}}^k\right)^{-1} \boldsymbol{\Sigma}_{\boldsymbol{\xi}\dot{\boldsymbol{\xi}}}^k. \tag{6}$$

The dynamical system with GMR is given by:

$$\dot{\hat{\boldsymbol{\xi}}} = \sum_{k=1}^{K} h^k(\boldsymbol{\xi}) \left(\boldsymbol{\Sigma}_{\dot{\boldsymbol{\xi}}\boldsymbol{\xi}}^k \left(\boldsymbol{\Sigma}_{\boldsymbol{\xi}}^k\right)^{-1} \left(\boldsymbol{\xi} - \boldsymbol{\mu}_{\boldsymbol{\xi}}^k\right) + \boldsymbol{\mu}_{\dot{\boldsymbol{\xi}}}^k\right) \tag{7}$$

$$= \sum_{k=1}^{K} h^k(\boldsymbol{\xi}) \left(\mathbf{A}^k \left(\boldsymbol{\xi} - \boldsymbol{\mu}_{\boldsymbol{\xi}}^k\right) + \mathbf{b}^k\right), \tag{8}$$

where $\mathbf{A}^k = \boldsymbol{\Sigma}_{\dot{\boldsymbol{\xi}}\boldsymbol{\xi}}^k \left(\boldsymbol{\Sigma}_{\boldsymbol{\xi}}^k\right)^{-1}$, $\mathbf{b}^k = \boldsymbol{\mu}_{\dot{\boldsymbol{\xi}}}^k$ and $h^k(\boldsymbol{\xi})$ are described by (4). Khansari-Zadeh et al. [2010] propose a learning algorithm called Stable Estimator of Dynamical Systems (SEDS), which estimates the parameters $\boldsymbol{\theta}^k = \left\{h^k, \boldsymbol{\mu}^k, \boldsymbol{\Sigma}^k\right\}$ $(k = 1..K)$ of the model (8) with the constraints

$$\begin{cases} \mathbf{b}^k = \mathbf{A}^k \left(\boldsymbol{\mu}_{\boldsymbol{\xi}}^k - \boldsymbol{\xi}_g^{\mathrm{T}}\right) \\ \dfrac{1}{2}\left(\mathbf{A}^k + \left(\mathbf{A}^k\right)^{\mathrm{T}}\right) : \text{negative definite} \\ \boldsymbol{\Sigma}_{\boldsymbol{\xi}}^k : \text{positive definite} \qquad \forall k \in 1..K \\ 0 < \pi^k \leq 1 \\ \displaystyle\sum_{k=1}^{K} \pi^k = 1, \end{cases} \tag{9}$$

where $\boldsymbol{\xi}_g$ is the attractor of the dynamical system. The constraints ensure the global stability of the system and preserve the nature of the GMM.

3. CATCHING MOVING OBJECT

3.1 Catching moving object with consideration of the shock

In this section we apply the GMM and GMR to learn and generate a movements to catch a moving object. Learning to catch a flying ball without consideration of

impulse exchange during contact is presented by Kim et al. [2010]. The additional challenge emerging with the consideration of the impulse exchange is that not only the relative position error but simultaneously the relative velocity error between the catcher and the object must be controlled with appropriate coordination.

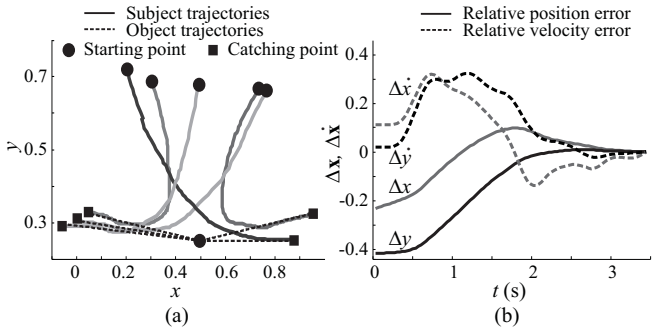

Fig. 1. Examples of catching moving object. (a) object trajectories (dashed line) and subject trajectories (solid line) and (b) relative position error (solid line) and relative velocity error (dashed line) between object and subject

To simulate the catching process, a subject is asked to move a computer mouse to catch a moving point on a screen and continue the movement after catching the point. Because of this kind of simulation, we do not use any unit for the variables in the next figures and results. Figure 1 presents some examples of the movement of the catcher and the object and shows the same characteristics of the movement as analyzed by Kajikawa et al. [1999], in which the catcher first moves to the object with a seemly straight line, then accelerates to reduce the velocity error, finally catches the object and continues the movement before stopping.

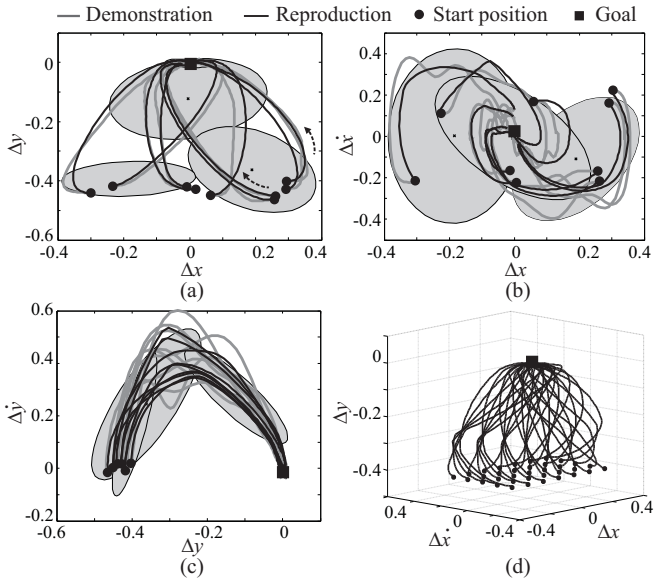

Fig. 2. Performance of GMM and GMR approximated by 4 Gaussians. (a) - (c) Reproduction of the demonstrated trajectories, (d) Stream lines of the generated trajectories starting with $\Delta y = -0.4$, $\Delta\dot{y} = 0$ and a meshgrid of Δx and $\Delta\dot{x}$

In order to learn both, the relative position $\mathbf{x} = [\Delta x, \Delta y]^{\mathrm{T}}$ and velocity $\dot{\mathbf{x}} = [\Delta\dot{x}, \Delta\dot{y}]^{\mathrm{T}}$ between the catcher and the object, the state vector is defined as $\xi = [\mathbf{x}, \dot{\mathbf{x}}]^{\mathrm{T}}$. The relative position $\mathbf{x}$ is computed from the object pose $\mathbf{x}_{\mathrm{o}}$ and the robot pose $\mathbf{x}_{\mathrm{r}}$ according to

$$\mathbf{x} = \mathbf{x}_{\mathrm{o}} - \mathbf{x}_{\mathrm{r}}. \tag{10}$$

The GMM and GMR are used to predict the derivative $\hat{\dot{\xi}} = \begin{bmatrix} \dot{\mathbf{x}} \\ \ddot{\mathbf{x}} \end{bmatrix}$ based on relation (8).

Figures 2 (a) to (c) compare the demonstrated and reproduced trajectories. They illustrate the good reproduction the learning trajectories by the GMM. The gray ellipses represent the approximated Gaussian functions, where the mean and the covariance of the Gaussian are indicated by the center and the semiaxes of the ellipse. By choosing the state variable as described above, the GMM allows to learn and reproduce a variety of different trajectories with identical initial positions but approaching the goal from different directions. This is shown by the dashed arrows in fig. 2 (a). The different directions of approach account for altered movement directions direction of the object. Figure 2 (d) illustrates the stability and the generalization ability of the dynamical model with varying initial states located on a meshgrid. All trajectories converge to the goal with a similar motion pattern, even if they start from the locations within state space, which are outside the demonstration set. The result indicate excellent generalization capabilities in terms interpolation and extrapolation of the imitation sheme.

The generalization ability of the system is shown from another perspective in fig. 3 (a) and (b). Both figures capture the path of the object as well as the catcher (fig. (a)) as well as the time evolution of the motion velocity (fig. (b)). An important characteristic of catching motions observed with human's can be clearly seen. The catching time is shorter when the object moves faster. Moreover, the system can be used for catching objects at rest, even though this scenario was not part of the demonstrations. Figure 3 (c) - (f) illustrate the robustness of the system to perturbations. In particular, the object suddenly changes the direction of motion (fig. (c) and (d)) or moves on an plane with positive (inclined downward plane), negative (with friction) or large negative (inclined upward plane) acceleration (fig. (e) and (f)).

3.2 Catching moving object with implicit time

Kim et al. [2010] propose a temporal scaling to modify the execution speed:

$$\lambda_{k+1} = \lambda_k + k_P \left(\hat{T}_k - T\right) + k_D \left(\hat{T}_k - \hat{T}_{k-1}\right), \tag{11}$$
$$\lambda_0 = 1,$$

where T is the desired duration of the maneouver, $\hat{T}_k$ is the estimated duration including the period elapsed betweenthe beginning of the movement and the current time step t_k as well as time between t_k and the convergence to the goal predicted by using the model in (8).

The velocity mean $\boldsymbol{\mu}_{\dot{\xi}}^{k}$ and the covariance $\boldsymbol{\Sigma}_{\dot{\xi}\dot{\xi}}^{k}$ are multiplied by the factor λ and thus the GMM system becomes:

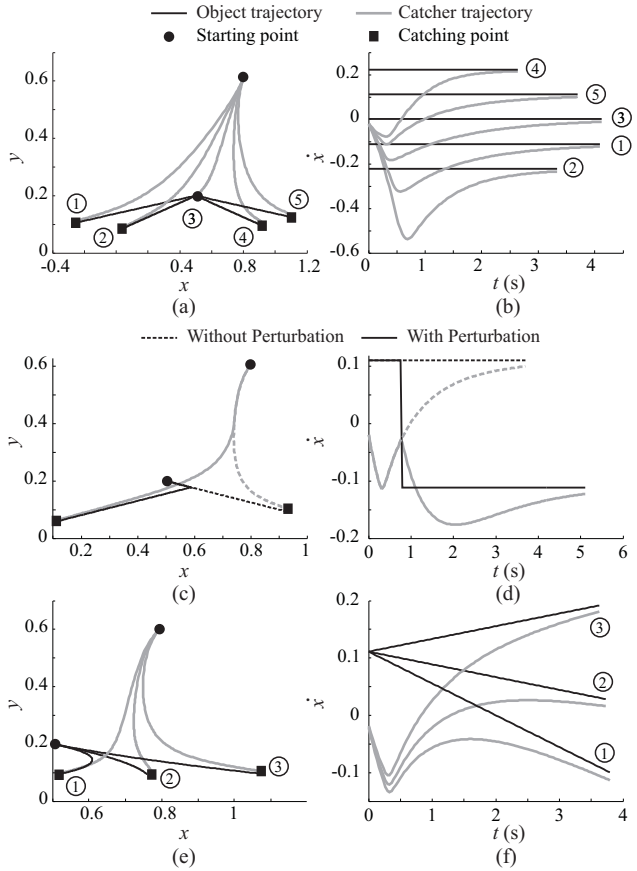

Fig. 3. Robustness to perturbations. (a) - (b) robustness to different configurations, (c) - (d) robustness to suddenly change of the direction, (e) - (f) robustness to velocity change

$$\begin{aligned}\widetilde{\dot{\hat{\boldsymbol{\xi}}}} &= \sum_{k=1}^{K} h^k\left(\boldsymbol{\xi}\right)\left(\lambda\boldsymbol{\Sigma}^k_{\dot{\boldsymbol{\xi}}\boldsymbol{\xi}}\left(\boldsymbol{\Sigma}^k_{\boldsymbol{\xi}}\right)^{-1}\left(\boldsymbol{\xi}-\boldsymbol{\mu}^k_{\boldsymbol{\xi}}\right)+\lambda\boldsymbol{\mu}^k_{\dot{\boldsymbol{\xi}}}\right)\\ &= \lambda\sum_{k=1}^{K} h^k\left(\boldsymbol{\xi}\right)\left(\mathbf{A}^k\left(\boldsymbol{\xi}-\boldsymbol{\mu}^k_{\boldsymbol{\xi}}\right)+\mathbf{b}^k\right)=\lambda\dot{\hat{\boldsymbol{\xi}}}. \qquad (12)\end{aligned}$$

In equation (12) the original predicted velocity from (8) is multiplied with λ and allows to speed up or slow down the motion. The scaling equation (11) with properly tuned parameters k_P and k_D can be interpreted as a PID controller with the maneouver execution time as the control variable.

The verification of this timing controller is shown in figure 4. The dashed light lines in fig. 4 (a) and (b) present the unmodified trajectoties, where the object is caught at 2.3 s. The modified system is applied and set to catch the object at the time point of 2.5 s, 3.0 s and 3.5 s. Figure 4 (b) illustrates that the object is being caught at 2.47 s, 3.00 s and 3.47 s, respectively.

According to Kajikawa et al. [1999], the catching position is determined based on the prediction of the object movement, but independent of the object velocity. In this work, the catching point is selected as to maximize the time left for catching the object after the approaching maneouver. Based on this point, the system adjusts the velocity and the timing, so that the object is caught either at the desired time or rather at the desired position. Figure 4 (c) and (d) show the archievement of using the system to catch the object at the desired position. In the scenario the catcher and the object always start from a same positions. However, the object moves with different velocities.

Figure 5 depicts the control architecture used in this work. The camera system detects the object and provides the position and velocity of the object. The relative position and velocity error between the robot and the object are calculated and defined as the input of the GMM. GMR estimates the model output and computes the velocity and acceleration error. The catching point is chosen and the temporal scaling of the timing controller is adjusted. The desired velocity and acceleration of the robot are finally determined and handed to the robot low-level controller. GMM is asymptotically stable when it is approximated with SEDS as proved by Khansari-Zadeh et al. [2010]. The whole system is thus globally stable for positive values for the proportional gain K.

4. EXPERIMENTS

In this section we perform two experiments with the five degree of freedom Katana robot arm, The experimental system is depicted in fig. 6 (a). The 3D Camboard Nano PMD camera with a field of view of $90° \times 68°$, a framerate of 90 fps and a solution of 160×120 pixels provides not only the intensity but also the depth images and is used to detect the object. The object is a pingpong ball. In order to detect the pingpong ball, we use the same method as presented in Phung et al. [2011], in which a pingpong ball is segmented from the depth image by region growing algorithm (Ballard and Brown [1982]). The background is removed from the depth image, the region in the image, which is most similar to a circle, is defined as the ping pong ball to be caught. The transformation between the camera coordinate frame and robot coordinate frame is obtained from prior calibration. The teacher demonstrates the desired trajectory by moving the robot endeffector to

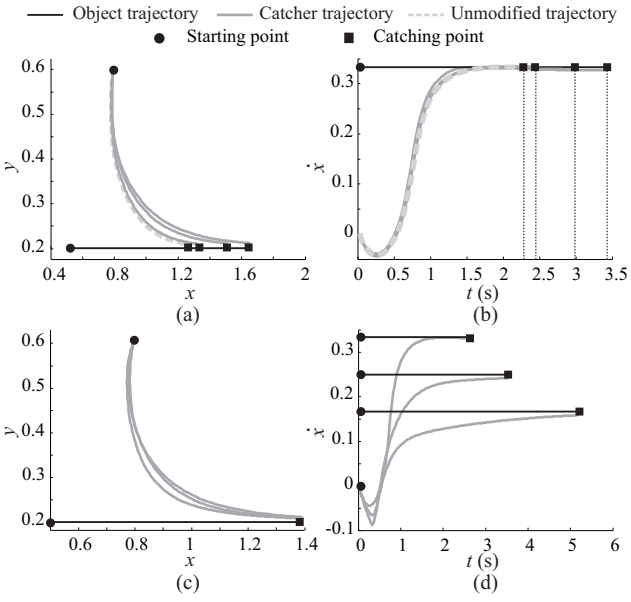

Fig. 4. GMM with timing controller. (a) - (b) catching time control, (c) - (d) catching point control

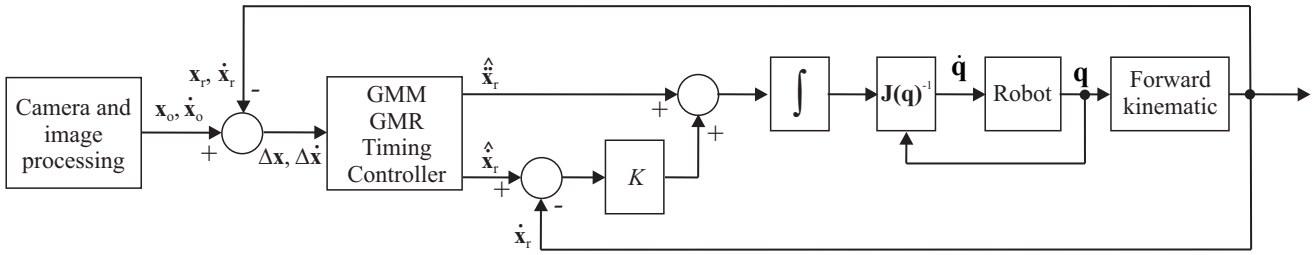

Fig. 5. Block scheme of control architechture

catch a pingpong ball running on a downward oriented plane. Figure 6 (b) depicts the demonstrated robot endeffector and the object trajectories in the robot coordinate system. Note, that in these experiments we also perform the "rendezvous" task but not catching task and the task is considered to be finished when the differences between the robot and the object position as well as velocity are smaller than a threshold value. The position of the object is defined as the topmost point of the pingpong ball to avoid the collision between the robot end effector and the object during the movement.

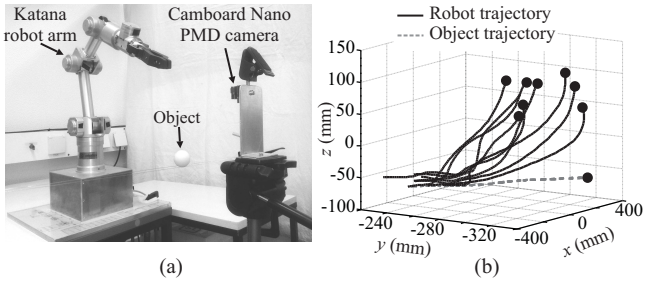

Fig. 6. Experimental system (a) and demonstrated robot (solid dark lines) and object (dashed light line) trajectories (b)

In the first experiment, the object is mounted on a stick and moved in a constant direction. The initial position of the robot and the object are equivalent to the initial positions in the demonstration. Figure 7 (c) shows the trajectory of the robot and object expressed in the robot coordinate frame, while fig. 7 (a) and (b) show the position and velocity of the robot (solid lines) together with the object (dashed lines) in space. Although the velocity signal of the object is neither smooth nor exactly constant and corrupted by measurement noise, the robot also accomplishes the task and aligns its motion in parallel with the object trajectory. The relative velocity between the robot and the object is reduced as far as possible. The variables of the GMM shown in fig. 7 (d) converge to zero and present a stable system. The remaining errors are attributed to to friction and the time delay in the communication of commands to the robot.

The second experiment illustrates the robustness of the system against sudden changes in the object'S direction of motion as seen in fig. 8 (a) - (c). The velocity component in x-direction varies the sign at the time instant of 5 seconds. The robot and the object both start from the positions, which are completely differ from the initial positions in the demonstrations. Nevertheless, the robot aproaches the object during these first 5 seconds in spite of the

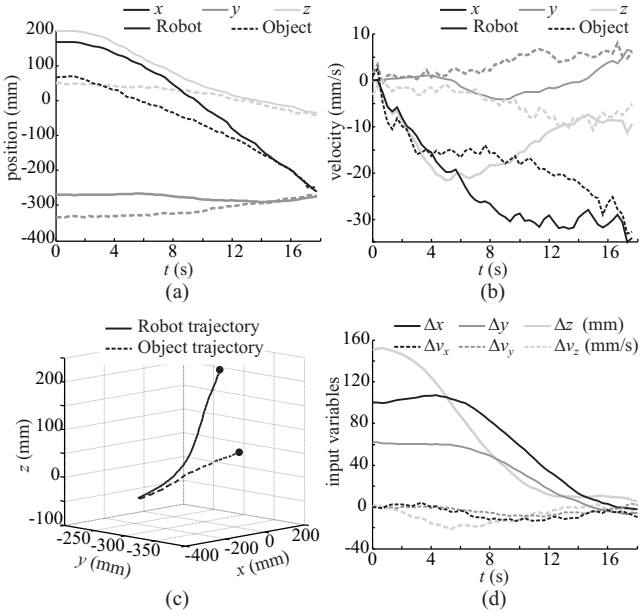

Fig. 7. Katana robot arm catching moving object with consideration of the damaging impact factor

previously unseen initial configuration as shown in fig. 8. Then it adopts to the altered object movement. The GMM generates the correct outputs to let the robot accomplish the catching task. It can be seen again in fig. 8 (d), that all variables of the GMM converge to the goal.

5. CONCLUSION

In this paper we present a learning approach using Gaussian mixture model together with Gaussian mixture regression to enable a robot to catch a moving object. The concept reduces the impulse exchange during contact phase by synchronizing the velocity vectors of the catcher and the object. A temporal scaling enables the adjustment of the finite catching time instead of actual the catching point along to the estimated object trajectory. This way the proposed concept can account for restrictions imposed e.g. by the robot workspace. Both, the simulated and the experimental results with a five degree of freedom Katana robot arm prove the reproduction and the generalization ability as well as the stability and robustness of the system with respect to untrained szenarios and sudden changes in the object trajectory.

The GMM models the relative position and velocity between the object and the robot with respect to a third eye-to-hand camera coordinate frame of reference. The GMR generates the relative velocity and acceleration in the same

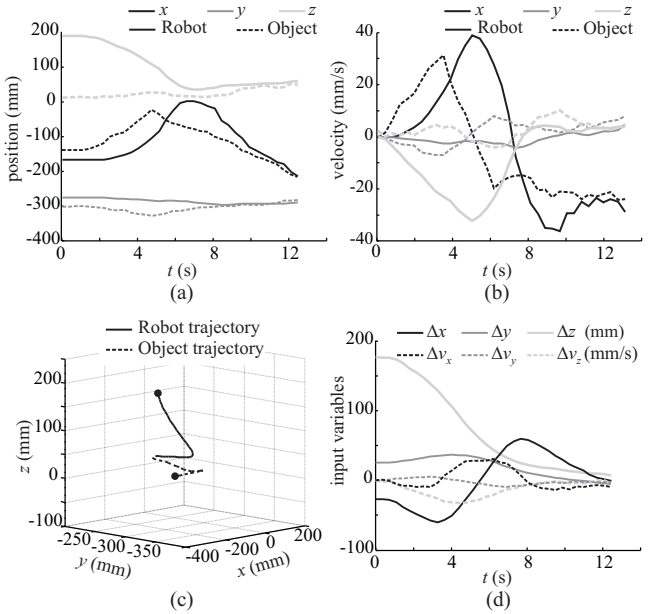

Fig. 8. Robustness of the system against a suddenly change of movement direction of the object

frame. The quantities have then to be transformed into the robot coordinate frame. The author's future works aim at the extension of the concept to an eye-in-hand camera , which directly measures the object movements in a robot centered frame of reference. This is expected to yield improved precision in conjunction with visual servoing controllers based on the difference between the actual view and the desired view of the object. Next, the catching tool with a mounted camera may be detached from the actual robot during the instruction phase. This presumably facilitates the demonstrations. The instructor can focus on guiding the camera without the need of paying attention to the configuration of a backdrivable arm.

ACKNOWLEDGEMENTS

We gratefully acknowledge the financial support for this project by Karl Kolle Foundation.

REFERENCES

J. Aleotti and S. Caselli. Robust trajectory learning and approximation for robot programming by demonstration. *Robotics and Autonomous System*, 54:409–413, 2006.

D. H. Ballard and C. M. Brown. *Computer Vision*. Computer Vision. Prentice Hall, 1982.

S. Calinon, F. Guenter, and A. Billard. On learning and representing and generalizing and a task and in a humanoid and robot. *IEEE Transactions on Systems, Man and Cybernetics PART B: Special issue on robot learning by observation, demonstration and imitation*, 37:286–298, 2007.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society, series B*, 39:1–38, 1977.

U. Frese, B. Baeuml, G. Schreiber, I. Schaefer, M. Haehnle, and G. Hirzinger. Off-the-shelf vision for a robotic ball catcher. In *In Proceedings of the IEEE/RSJ Intl. Conference on Intelligent Robots and Systems Oktober 2001, Maui*, pages 1623–1629, 2001.

E. Gribovskaya, S.M. Khansari-Zadeh, and Aude Billard. Learning nonlinear multivariate dynamics of motion in robotic manipulators. *International Journal of Robotics Research*, 30:80–117, 2011.

H. Hoffmann, P. Pastor, and S. Schaal. Dynamic movement primitives for movement generation motivated by convergent force fields in frog. In *adaptive motion of animals and machines (amam)*, 2008.

S. Kajikawa, M. Saito, K. Ohba, and H. Inooka. Analysis of human arm movement for catching a moving object. In *In IEEE International Conference on Systems, Man, and Cybernetics*, pages 698–703, 1999.

S. M. Khansari-Zadeh and Aude Billard. Bm: An iterative algorithm to learn stable non-linear dynamical systems with gaussian mixture models. In *In Proceeding of the International Conference on Robotics and Automation*, pages 2381–2388, 2010.

S. M. Khansari-Zadeh, Aude, and Billard. Imitation learning of globally stable non-linear point-to-point robot motions using nonlinear programming. In *In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.

S. Kim, E. Gribovskaya, and A. Billard. Learning motion dynamics to catch a moving object. In *In IEEE International Conference on Humanoid Robots*, 2010.

J. Malzahn, A. S. Phung, and T. Bertram. A multi-link-flexible robot arm catching thrown balls. In *7th German Conference on Robotics; Proceedings of ROBOTIK 2012;*, pages 1–6, 2012.

A. Namiki and M. Ishikawa. Robotic catching using a direct mapping from visual information to motor command. In *Proc. IEEE Int. Conf. Robot. and Automat*, pages 2400–2405, 2003.

A. S. Phung, J. Malzahn, F. Hoffmann, and T. Bertram. Get out of the way - obstacle avoidance and learning by demonstration for manipulation. In *18th IFAC World Congress, Milano, Italy*, 2011.

M. Riley and C. G. Atkeson. Robot catching: Towards engaging human-humanoid interaction. *Journal Autonomous Robots*, 12:119–128, 2002.

S. Schaal, P. Mohajerian, and A.J. Ijspeert. Dynamics systems vs. optimal control a unifying view. *Progress in Brain Research*, 165:425–445, 2007.