Robotic Pre-Manipulation: Real-Time Polynomial Trajectory Control for Dynamic Object Interception with Minimum Jerk

Arjun Nagendran[1], Remo Pillat[1] and Robert Richardson[2]
[1]*Institute for Simulation and Training, University of Central Florida, 3100 Technology Pkwy, Orlando, FL 32826, USA*
[2]*School of Mechanical Engineering, University of Leeds, Leeds, LS2 9JT, UK*
{*arjun, rpillat*}*@cs.ucf.edu, r.c.richardson@leeds.ac.uk*

Keywords: Object Capture, Trajectory Planning, Minimum-Jerk.

Abstract: This paper presents a method for capturing a free-moving object in the presence of noise and uncertainty with respect to its estimated position and velocity. The approach is based on Hermite polynomials and involves matching the state-space parameters of the object and the end effector at the moment of contact. The method involves real-time re-planning of the robot trajectory whenever new estimates of the object's motion parameters are available. Continuity in position, velocity, and acceleration is preserved independently of the planning update rate and the resulting trajectories are characterized by low jerk. Compared to other methods that directly solve for higher-order polynomial coefficients, the proposed algorithm is computationally efficient and does not require a linear solver. Experimental results confirm the advantages of this method during real-time interception of a dynamically moving object with continuous velocity estimation and high-frequency re-planning.

1 INTRODUCTION

Robotic manipulation is an important problem in several application domains. A prerequisite for robotic manipulation is the ability to 'pick up' an object. For objects that are static in nature, this essentially reduces to a path planning problem for the end effector of the robotic manipulator. Unless the environment continuously changes, it is relatively straightforward to pre-plan a path for the end effector while ensuring smooth motions for the robotic arm. *Moving* objects present a special challenge, because they may have to be intercepted or grasped before they can be manipulated. Measurement uncertainties during path planning of the robotic manipulator can result in rapid changes in the motion profile of the system, thereby subjecting the mechanism to jerk (third derivative of position) and increasing wear on the system. Additionally, any mismatch in the velocities of the object and the manipulator at the time of impact could cause damage to the manipulator, captured object or both entities.

"*Minimum jerk trajectories are desirable for their similarity to human movements and help improve robot life-span by limiting robot vibrations* " (Piazzi and Visioli, 1997)

Traditionally, higher-order polynomials are used to minimize jerk during robotic motion, but are mostly pre-planned with a priori knowledge of the target coordinates. When used in the time domain, the coefficients of these curves can be varied to create a motion profile for the end effector of a manipulator. By choosing a polynomial with a sufficiently high degree, it is possible to preserve continuity of its second derivative, i.e. acceleration, thereby minimizing jerk. Evaluating these higher-degree polynomials can however be computationally expensive. This is of particular interest since the coefficients of these polynomials must be tuned so that the end effector motion profile adapts to the (varying) velocity of an incoming object in real-time and at potentially high update rates.

The contribution of this work is in achieving low-jerk motion profiles for robotic end effectors while ensuring minimum impact forces during interception and capture. This is achieved by continuously matching state-space parameters (position, velocity and acceleration) between the object and the end effector at the predicted rendezvous point. In contrast to existing methods, the presented method has a high tolerance for uncertainties and noise in the state space estimates of the approaching object while being computational efficient. The coefficients of a higher-order polynomials can be tuned in real-time and at high update rates to ensure the successful interception of dynamic objects. A video demonstration of this capture process can be viewed at http://youtu.be/yTx4a9UAMAM. This paper describes the following:

- Choosing suitable polynomial trajectories that are characterized by low-jerk and allow sufficient flexibility to alter position, velocity and acceleration while preserving continuity at every instant in time during real-time capture.
- Re-computing the polynomial trajectories at every instant in response to incoming velocity estimates at a sampling frequency of 200 Hz.
- Reducing jerk during the entire process of interception, with the additional capability of extending the same acceleration profile for post-capture (deceleration).
- Experimental testing and results of the method during real-time interception of a dynamically moving object with continuous velocity estimation in the presence of noise.

2 BACKGROUND

During robotic interception and capture, a difference in velocity at the grasping instant may result in slippage or impact which can cause damage to the object or even damage to the manipulator (Lin et al., 1989). Traditional methods of solving this velocity matching problem involve predicting the position of an incoming object and positioning the end effector at the computed interception point to capture the object (Borg et al., 2002; Bauml et al., 2010). Several path-planning techniques exist to achieve this interception point including the use of polynomials in general as described in (Croft et al., 1998; Papadopoulos et al., 2005) and cubic splines in specific (Gasparetto and Zanotto, 2008; Wang and Horng, 1990). Many of these techniques use a one-shot design approach for the polynomials, assuming a priori knowledge of the incoming object such as constant acceleration and velocity (Campos et al., 2006) and resulting in pre-planned trajectories for robotic interception. While these methods are suitable in constrained environments with repetitive tasks, the velocity of the incoming object may change in unconstrained environments. The capture of such objects presents a challenge due to a continuously changing predicted interception point. Additionally, with purely predictive interception systems, impact and bounce are issues that must be addressed. High speed robotics (Namiki et al., 2003; Imai et al., 2004) overcomes the need for prediction by simply reacting to the changing position estimates of an object at a very high frequency. However, no consideration of smoothness or jerk reduction is possible with such systems.

Several researchers have previously addressed the need for velocity matching. Nelson et.al. (Nelson et al., 1995) emphasize in their work the need for fast and stable transitions from non-contact states to contact states, with minimum impact and bounce. An approach where force and vision feedback are simultaneously used to minimize the impact forces when a stiff manipulator makes contact with a stiff surface is presented. In effect, the static surface can be thought of as having zero velocity and the manipulator is forced to achieve minimum velocity just before contact. In (Lin et al., 1989), a coarse tuning algorithm which drives the end effector into the neighborhood of the object in minimum time, and a fine tuning algorithm which ensures zero relative velocity and acceleration at the time of grasping is presented. A sensor based system employing the above method is simulated for a two-degree of freedom (2DOF) planar robot at a sampling rate of 20 Hz.

The authors of (Kövecses et al., 1999) describe the equations of the robotic end effector and object independently before the interception phase. According to the authors, the dynamics are dependent on the velocities and angular velocities at the instant of contact. It is proved that when these components are zero, there are no impulsive forces acting on the system, resulting in a smooth capture. (Zhang and Buehler, 1994) define a smooth grasp as one in which the position, velocity and acceleration of the robot end effector matches that of the moving object at the point of contact.

While trying to achieve a certain end effector velocity, it is important to be able to control its trajectory. A trajectory defines the position, velocity and accelerations of the end effector in its task space coordinates. Confining the trajectory of the end effector to be in line with the path of the object can significantly help minimize the mismatch in velocities between the two. Commonly used trajectories such as a linear trajectory, with parabolic blends are subject to high jerk, due to sudden changes in acceleration. The also require the application of an instantaneous change in force by the actuator which is not practical. Other methods of trajectory generation include third- and fifth-order polynomials that describe the end effector path in joint space; fifth-order polynomials have the added advantage that they ensure low jerk values.

A third-order spline interpolation based trajectory planning method to minimize the instantaneous velocity change during bipedal robot foot contact is proposed in (Tang et al., 2003). Simulation results reveal that the trajectory obtained helps reduce impact. A method for generating smooth jerk-bounded trajectories is developed in (Macfarlane and Croft, 2003). Jerk limitation results in improved tracking and reduced wear on the robot. The method used involves the concatenation of fifth-order polynomials to generate a smooth trajectory between two way points.

The above methods require accurate and quick control of the end effector to effectively match the velocity of a moving object. The performance is also dependent on the actuator limits with respect to its ability to apply quick-changing forces. In keeping with these, it is required to efficiently design polynomial trajectories that are adaptive to changes in parametric conditions of objects being captured in real-time. Additionally, they must cope with changing parametric estimates obtained from sensor data that are subject to noise and uncertainties. This problem forms the focus of this work.

3 FORMAL PROBLEM DEFINITION

During robotic manipulation, a difference in velocities at the instant of grasping can result in impact, which can cause the object being captured to bounce off the end effector, and also cause damage to the object or end effector. The effect of the difference in velocities must therefore be understood, and a method to minimize this mismatch must be developed to ensure a successful grasp at the time of contact.

3.1 Velocity Matching and Impact: Influence on Kinetic Energy

The velocity matching problem along a single axis is illustrated in Figure 1. A mass M (kg) approaches the end effector of a serial-link robotic arm with a velocity V_{mass} (m/s). The end effector cannot move beyond its inner workspace limit D_{lim} (m). The mass is therefore required to be decelerated over a distance D_{stop} (m), a method for which was first described in (Nagendran et al., 2007) and improved upon in (Nagendran et al., 2011). From the known velocity of the incoming mass, an intercept point for the end effector and the object is computed (object travels a distance $D_{intercept}$ (m) before impact). The distance from the initial position of the end effector (its outer workspace limit D (m) as measured from the base coordinate frame) to the intercept point is given as D_{eff} (m). The process of velocity matching requires some acceleration of the manipulator so that the velocity of the end effector V_{eff} (m/s) matches V_{mass} (m/s) over this distance D_{eff} (m). Note that the differences in effective distances covered by the mass and the end effector from the start of the velocity matching process result in varied times over which the same velocities need to be achieved.

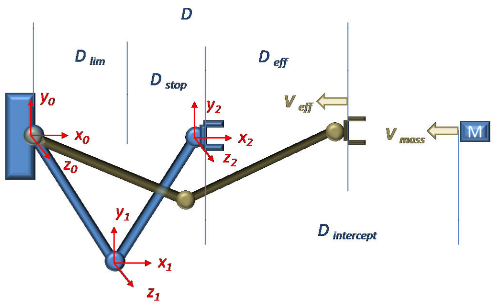


Figure 1: Illustration of the velocity matching problem along a single axis

When impact occurs, a portion of the kinetic energy of the two bodies is lost as thermal energy. According to Newtons experimental law (of impact / restitution), the loss of kinetic energy after impact of two straight point masses m_1 and m_2 with initial velocities v_1 and v_2 can be determined using

$$L_{ke} = \frac{1}{2}\frac{m_1.m_2}{m_1+m_2}(v_1-v_2)^2(1-{k_i}^2) \qquad (1)$$

where k_i is the coefficient of impact and is dependent on the material properties of m_1 and m_2. It is evident from Equation (1) that by minimizing the mismatch in the velocities of the two objects, the impact and hence the amount of kinetic energy lost can be minimized. The impact therefore serves as a quantitative measure of the performance of the velocity match in the experimental trials discussed in Section 6.

3.2 Constraints for Successful Velocity Matching

Let T be the time instant at which contact between the end effector and the object occurs and t be the continuous time axis.

- To ensure low impact during capture (and prevent bounce), the position, velocity and acceleration of the end effector must match that of the object at the time of capture, i.e. $e_{obj}(T)$ = $\dot{e_{obj}}(T)$ = $\ddot{e_{obj}}(T)$ (Zhang and Buehler, 1994), where e_{obj} represents the error between object and end effector state variables.
- Additionally, there must be no collision between the object and the end effector prior to the time taken by the object to reach the capture point.
- Due to inherent actuator dynamics, there will be errors between the desired and actual positions of

the end effector (e_{eff}). Ideally, these must be minimized during the process i.e. $\forall t \leq T$, $e_{eff} \neq 0$.

- The objective is to generate a position demand for the end effector so as to minimize $e_{obj}(T)$, $e_{\dot{obj}}(T)$ and $e_{\ddot{obj}}(T)$ within actuator limits. This determines the time required to match the object's position, velocity and acceleration and also the coordinate of the interception point in the workspace of the manipulator.

4 POLYNOMIAL TRAJECTORIES

Previously published methods in the area of robotic capture / interception generally assume a predictable incoming object velocity (constant motion, deceleration etc.) that serves as a basis for computing an interception point during the maneuver. However, real-world objects may not adhere to this assumption. Air resistance, surface friction, and energy losses in kinematic components can all contribute to a variable velocity. If the approaching object has a propulsion system, any prediction of its future velocity is subject to inaccuracy. Thus, adjustments to the end effector trajectory need to be made continuously for successful interception of the incoming object.

Ideally, the trajectory of the end effector can be represented through a piecewise 5th-order polynomial to ensure a minimum jerk trajectory. The constraints in order to perform velocity matching using this trajectory include specifying the position, velocity and acceleration at the initial and final positions of the end effector. This can be done using quadratic or cubic functions. However, since we require the acceleration profile to be a third-order polynomial in keeping with a minimum jerk trajectory, the desired trajectory is represented as a 5th-order polynomial. Solving for their coefficients in real-time is computationally expensive. By using the basis functions of Hermite spline, the same solution can be found more efficiently, thus allowing high-frequency re-planning of the end effector motion profile. Please see Section 4.2 for a comparison of computational efficiency between the two methods.

4.1 Hermite Splines

Hermite polynomials are a specific set of orthogonal basis functions whose linear combination spans the space of all polynomials of a certain degree. The splines are a piecewise concatenation of these polynomials, each of which is in the Hermite form. Depending on the degree of the polynomials, continuity in the derivatives will be preserved at the connecting knot points. For the purposes of this work, quintic Hermite splines will be considered, so minimum jerk trajectories can be enforced. A quintic Hermite polynomial is uniquely defined by a start and ending point and the first two derivatives of the function at these points.

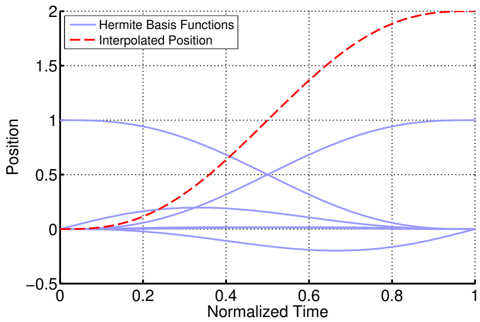


Figure 2: General example of quintic Hermite polynomial interpolation. The *Interpolated Position* is a linear combination of the *Hermite Basis Functions*.

For the normalized domain $t \in [0,1]$, the quintic Hermite basis functions $h(t)$ take the following form:

$$\begin{aligned}
h_0(t) &= -6t^5 + 15t^4 - 10t^3 + 1 \\
h_1(t) &= -3t^5 + 8t^4 - 6t^3 + t \\
h_2(t) &= -0.5t^5 + 1.5t^4 - 1.5t^3 + 0.5t^2 \\
h_3(t) &= 0.5t^5 - t^4 + 0.5t^3 \\
h_4(t) &= -3t^5 + 7t^4 - 4t^3 \\
h_5(t) &= 6t^5 - 15t^4 + 10t^3 \qquad (2)
\end{aligned}$$

In the general case, the start and end time of a trajectory segment are denoted as t_0 and t_f . x_0, v_0 and a_0 define the initial position, velocity, and acceleration at t_0, while x_f, v_f and a_f denote their final values at t_f.

For a given time $t_h \in [t_0, t_f]$, the corresponding normalized time in the $[0,1]$ domain can be found through:

$$t = \frac{t_h - t_0}{\Delta t} \qquad (3)$$

Here $\Delta t = t_f - t_0$ is the conversion factor between real and normalized time, but can also be used to normalize and de-normalize velocity and acceleration values. The actual position x and velocity v at this normalized time t can then be calculated. For notational simplicity it is assumed that p, v, h, and $\dot{h}$ are all functions of t.

$$\begin{aligned}
x &= x_0 h_0 + v_0 h_1 + a_0 h_2 + a_f h_3 + v_f h_4 + x_f h_5 \\
v_n &= x_0 \dot{h}_0 + v_0 \dot{h}_1 + a_0 \dot{h}_2 + a_f \dot{h}_3 + v_f \dot{h}_4 + x_f \dot{h}_5 \\
v &= v_n / \Delta t \qquad (4)
\end{aligned}$$

The derivatives $\dot{h}_0 \ldots \dot{h}_5$ can be found analytically. Similar expressions to Equation (4) can be derived for the end effector acceleration and jerk.

4.2 Computational Efficiency of Hermite Solution

Looking at Equation (4), it becomes immediately clear that the calculation of a demand position and demand velocity for any normalized time t involves nothing more than a fixed number of multiplications and additions. Furthermore, the computational complexity is constant regardless of changing start and end conditions. In the case of the pure fifth-order polynomial solution, a new set of coefficients would have to be calculated each time one of the variables $x_0, x_f, v_0, v_f, a_0, a_f$ changes.

To quantify the computational efficiency of the Hermite-spline-based solution, a test program was written in Matlab. Random vectors with 1 million elements were generated for the $t_0, t_f, x_0, x_f, v_0, v_f, a_0, a_f$ parameters. There is an additional vector t with the same number of elements that represents a random time sample between t_0 and t_f that is evaluated. Two different methods for calculating the end effector trajectory were implemented:

1. *Linear Solver*: The coefficients of the fifth-order polynomial can be found by solving a system of six linear equations. This method uses Matlab's optimized 'linsolve' function that is based on LU-decomposition. Using these coefficients, position, velocity, acceleration, and jerk can be evaluated for time instants t.

2. *Hermite-based Solver*: Based on Equation (4) and the corresponding calculations for acceleration and jerk, the end effector state for time instants t can be directly identified.

Both solvers were executed on each of the 1 million test cases and their respective runtimes recorded. It should be noted that both solutions result in precisely the same trajectory solutions, but their runtime differs. On a Core i7 processor with an operating frequency of 3.2 GHz, the average time required per iteration for solver (1) was 0.085 ms, whereas the Hermite-based solver (2) executes in 0.017 ms on average. The Hermite splines thus provide a speedup of almost 5 times over the linear solver.

It should be pointed out that the computational advantage is a requirement given the constantly changing start and end conditions due to noisy state estimates of the approaching object. If a continuous estimation and adjustment of the trajectory is desired, then the Hermite spline solution lends itself as an implementation of choice. Each time a new estimate of the approaching object's position and velocity is available, the current position, velocity, and acceleration of the actuator can be used as initial conditions for a new Hermite polynomial.

In addition, no linear solver is required for the Hermite solution, so implementation can be simplified if executing on resource-constrained embedded hardware.

4.3 Analysis: Hermite Splines for Velocity Matching

The following section discusses the effect of continually changing velocities on the motion profile of an end effector computed using Hermite splines. By adaptively changing the motion profile of the end effector to incoming velocity estimates, a smooth motion profile for the end effector can be achieved. As an added advantage, the velocity matching controller does not require these updates at a fixed frequency; a new estimate of the object's velocity can be used as soon as it is available thereby making the controller robust to data loss and sensor noise.

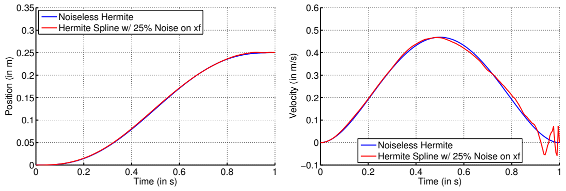


Figure 3: Achieving new motion profiles using Hermite splines in the presence of noise

The quintic Hermite basis functions are first computed for a given instant in time as described by Equation (2). Equation (4) is then used to compute a demand position and demand velocity for the end effector at every instant in time. Figure 4 illustrates the process graphically. An initial estimate of the motion profile for the end effector is used to compute the hermite spline that achieves 0.225m in 0.5s. At t = 0.2s, a new estimate is acquired which requires the end effector to be positioned at 0.275m at t = 0.5s. The coefficients of the hermite spline are computed for the new estimates resulting in a new 'blended' hermite spline as seen in the figure. Notice the continuity in position and smooth changing velocity at t = 0.2s despite the jump in target coordinate (by 22%). The acceleration profile and jerk profiles also reflect these changes but are still within acceptable limits and show no large spikes. Figure 3 shows the position and velocity profiles generated in the presence of 25% Gaussian, zero-mean, additive noise on the end position estimate $x_f = 250$ mm. A new Hermite spline is estimated at each 0.05 s time step, starting from the current position, velocity, and acceleration. The resulting motion profile of the end effector is still smooth and

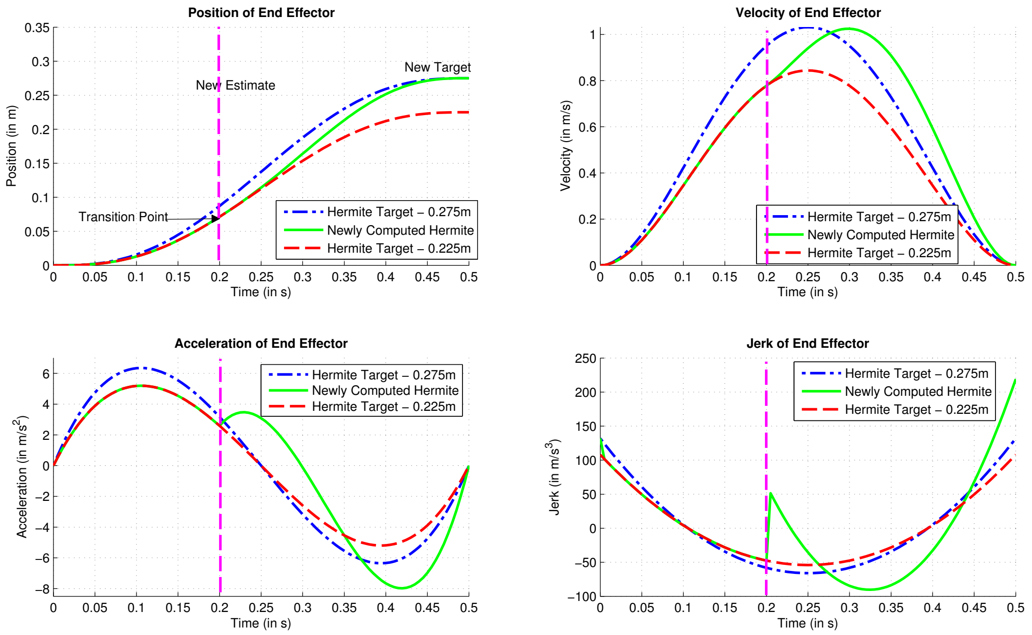


Figure 4: Motion profiles of end effector during hermite blending

can therefore be altered in real-time without causing large changes in acceleration or jerk during the process. The velocity spikes at the end of the trajectory are due to the relatively drastic changes to x_f that need to occur in a shorter and shorter time span. To avoid this behavior in real experiments, it is recommended to cease the Hermite interpolation when the object is within a small band before the target.

5 EXPERIMENTAL SETUP

An overview of the experimental setup showing the hardware components (Table 1) is depicted in Figure 5. The mechanism used to illustrate the velocity matching controller in the experiments is a 1 degree-of-freedom (DOF) LinMot linear actuator. The LinMot is an electromagnetic device that can move a slider linearly without the use of screws or gears. Its position can be resolved with high accuracy within 0.05 mm. This is representative of the end effector position along a single axis (see Figure 1) that can be extended to serial link manipulators as desired. An uniaxial force sensor from Transducer Techniques is mounted at the end of the slider and used to record forces encountered when in object contact. Desired positions and velocities are commanded to the actuator at a constant sampling frequency of 200 Hz. On a lower level, this control mode circumvents any secondary motion planning, but the smoothness and feasibility of the input trajectory has to be ensured by the implemented motion controller.

Two separate carts were used as objects approaching the actuator. One of the carts can be remotely-driven at discrete constant velocities, while the other is free moving. The free-moving cart is rolled down a ramp towards the end effector and sometimes impeded using undulations as shown in Figure 5. One wheel shaft on each robot is rigidly connected to an incremental, quadrature optical encoder that produces 90 pulses per revolution. The encoder pulses are counted by an Arduino microcontroller board and wirelessly sent to the controlling computer via an XBee RF module. This information is used to compute the incoming velocity of the cart.

The velocity matching controller is implemented in Labview and sends a position and velocity command stream to the Linmot actuator over a CAN bus interface. The minimum-jerk trajectory is calculated at 200 Hz and the actual position and velocity on the currently active 5th order polynomial is sent to the Linmot actuator. The force sensor is read through a National Instruments data acquisition board at 200 Hz.

An ideal controller would execute the velocity matching process as illustrated in Figure 6. When the cart approaches the end effector, the linear actuator would execute a smooth trajectory to match the cart's position, velocity, and acceleration at the capture point. Once contact is established, the object can be decelerated in a predefined workspace such as that described in (Nagendran et al., 2011).

Table 1: Hardware Components

Image	Component Description	Function	Specifications
(A)	Linear Actuator (LinMot) PS01-23x160 stator with PL01-12x760/710 Slider	Implements end effector motion profile (using a closed-loop PID Controller)	Sampling Frequency: 200 Hz Max. Velocity: 3 m/s
(B, C)	Carts (VEX)	(a) Remote-controlled with discrete velocity (b) Free-Moving (rolled down a ramp)	(a) Constant Velocity with single transition (b) Continuously changing velocity
(D)	Optical Shaft Quadrature Encoder (VEX)	Measures velocity of incoming object / cart (for real-time Polynomial Tuning)	Sampling Frequency: Velocity-dependent Resolution: 90 ticks/revolution
(E)	Uni-axial Force Sensor MDB-25 (Transducer Techniques)	Measures impact of incoming object with end effector (for theoretical verification)	Sampling Frequency: 200 Hz Max. Force: 111.20N
(F)	PointGrey Flea2 FL2-03S2C-C	Captures final frames prior to impact (for data review)	Frame Rate: 140 fps Resolution: 640 x 160
(G)	Custom Photo-Diode Beam Breaker Circuit	Provides reference position and initial velocity estimate of incoming object	Sampling Frequency: 200 Hz Output Voltage: 2.2V – 4.5V

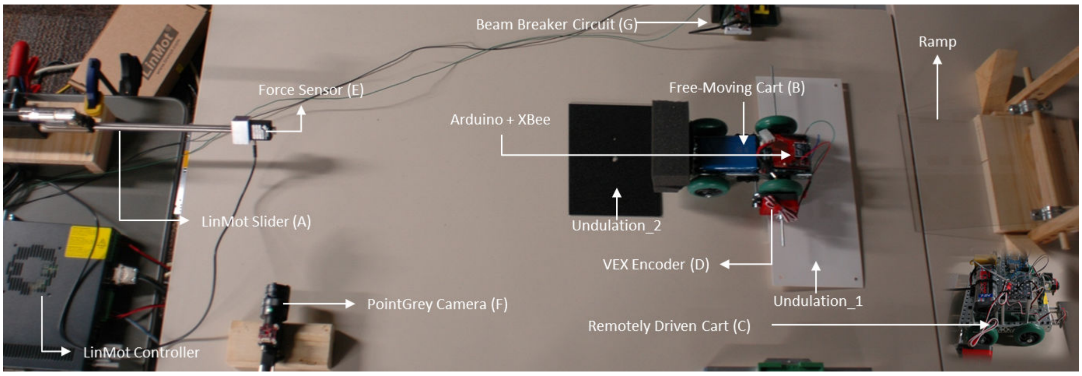


Figure 5: The experimental setup with linear actuator, adjustable ramp, and mobile robots

6 RESULTS

Several experimental trials were conducted to illustrate the repeatability and robustness of the controller to continually changing velocity estimates prior to interception. To illustrate the need for a continuously updated motion trajectory, an experiment that relies on a single measurement of the object's velocity is first described and analyzed. This is representative of a single-shot capture approach where an incoming object's velocity is known beforehand and planning occurs based on this apriori estimate. Experiments that involved constant velocity motion (from the remotely operated cart) and changing velocity motion (from the free moving cart, rolled down the ramp) were conducted. Results presented below focus on the free-moving scenario, since the constant velocity case is fairly trivial to handle.

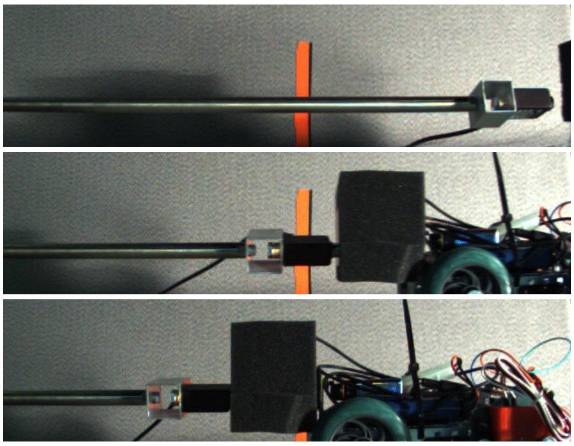


Figure 6: A typical velocity matching maneuver. The cart approaches the end effector in the topmost image, while the middle image shows the moment when contact is established. The cart is decelerated and comes to rest in the position shown in the bottom photo.

6.1 One-time velocity estimate

A single-shot approach is subject to failure in two instances i.e. the incoming object's velocity may either be (a) lower or (b) higher than originally estimated. In the former case, capture may never occur within the pre-defined distance while in the latter case, capture occurs prior to the pre-defined interception point, resulting in a high impact force. The latter case is considered for analysis in this subsection. An ideal point in the workspace of the LinMot is identified as the interception (capture) point. This, for instance, can be chosen depending on the portion of the workspace required for deceleration of the object. A beam-breaker circuit (with photo-diode) is used to estimate the velocity of the incoming cart and passed to the Hermite polynomial trajectory planner. Based on this received data, the time-to-impact is used to plan a minimum-jerk spline to catch the object at the 'target point'.

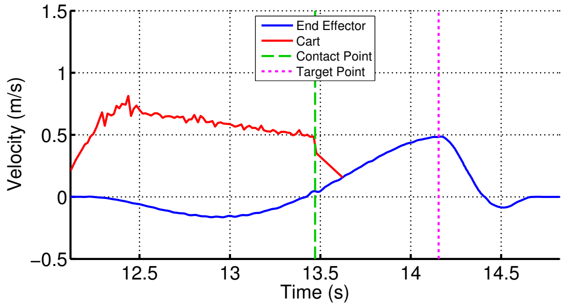


Figure 7: Velocity graphs reveal a mismatch due to an early contact between incoming object and end effector

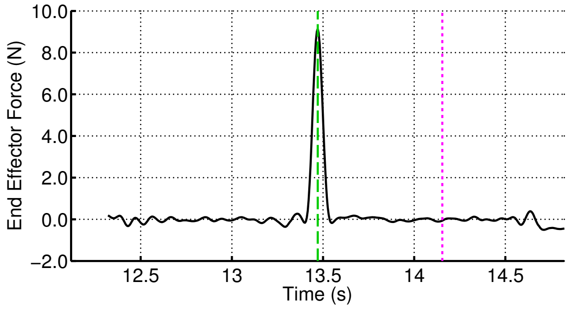


Figure 8: Corresponding force graph reveals a large impact force due to the mismatch in velocities

Figure 7 shows the velocity graphs of the incoming cart and the end effector during a failed interception scenario. The LinMot receives a one-time estimate of the cart's velocity, which is lower than the subsequent velocity achieved by the cart. .The planned motion is such that this velocity is achieved over a pre-defined distance of 0.225 m in the LinMot's workspace, with the time-to-impact calculated based on the distance of the cart. The cart's higher velocity results in interception at a co-ordinate well before the 0.225 m target is achieved, since the time to contact is much lesser than originally estimated. The result is a high impact force, as seen in Figure 8 due to the mismatch in velocities at the point of contact, and consequently a higher jerk due to the change in acceleration.

6.2 Continuous velocity estimate

In order to minimize impact and reduce jerk, it is required to continuously adapt the end effector's motion to match the incoming object's velocity. While doing so, it is required to ensure smooth motion, without discrete changes in acceleration that result in jerk. This is implemented with the Hermite Spline solution discussed in Section 4.3 at a frequency of 200 Hz. Several runs at different incoming velocities were conducted by varying the angle of the ramp that was used to launch the cart towards the end effector. Five separate runs were executed for each experimental condition and results are tabulated in Table 2. The mean and standard deviation for different metrics are shown and the important fields i.e velocity matching percentage and the impact force are indicated by shaded cells. In all conditions the velocities between linear actuator and cart were matched to within 7.5 % and the quality of the match is maintained through a range of different absolute velocity magnitudes. The force encountered on impact is on average less than 0.872 N, with conditions 1 and 3 showing even better results. These experiments verify the quality of the presented object capture method.

To further demonstrate how the presented method copes with uncertainty in the velocity estimates, several runs from *Condition 3* will be analyzed. In the experimental setup, two undulations of varying magnitude (Figure 5) were placed in the path of the incoming cart, causing it to unevenly decelerate as it approached the end effector. Velocity estimates in this case could also degrade due to skipped encoder counts when the undulations are traversed.

Several results from this run are shown in Figure 9 and analyzed in detail. To evaluate the quality of the velocity matching at this target location, velocities of end effector and cart against time are plotted in Figure 9a. At the impact time of 25.9 s, the linear actuator's velocity is 0.4135 m/s^2, while the cart's velocity is 0.436 m/s^2 - a velocity mismatch of approximately 5.2%. The positions of the cart and the linear actuator are displayed in Figure 9(b). Performance is further analysed by measuring the force encountered on impact of the cart. Figure 9(c) shows the value of the measured forces throughout the run. At time of impact, the force is evaluated to 0.428 N. Compare this with the significantly higher value described in the failure case above. Please note that, for visualization clarity, the force data has been filtered with a low pass Butterworth filter. It should be pointed out that these promising results are achieved in the presence of a constantly changing estimate for the time to impact (see Figure 9(d)).

Table 2: Results of adaptive velocity matching experiments. Five runs were executed in each experimental condition.

	Condition 1		Condition 2		Condition 3	
	mean	stdev	mean	stdev	mean	stdev
Contact Point (m)	0.196	0.018	0.179	0.008	0.199	0.018
Contact Linmot Velocity (m/s)	0.221	0.026	0.380	0.028	0.345	0.038
Contact Cart Velocity (m/s)	0.239	0.023	0.401	0.025	0.353	0.045
Velocity Match (%)	92.540	2.212	94.591	1.374	96.328	2.266
Impact Force (N)	0.537	0.233	0.872	0.106	0.413	0.094

7 DISCUSSION, CONCLUSION AND FUTURE WORK

In this work, a new approach to robotic interception of moving objects was proposed. The method relies on the use of higher order polynomials that exhibit low jerk properties for continuous real-time trajectory planning of robotic end effectors. In specific, the use of Hermite splines for catching moving objects in the presence of sensor noise was discussed. Results reveal a high accuracy of velocity matching prior to interception resulting in low impact and reduced jerk. Additionally, continuity of the end effector state-space variables (up to the second derivative) are maintained during the process and can be extended to the post-capture stage of robotic interception. Although Hermite splines have attractive properties (see Section 4.1), they have some drawbacks when used for generating trajectories:

- The start and end constraints are guaranteed to be fulfilled, but over and undershoot inside the interpolation interval can occur. This is problematic if the actuator has fixed motion constraints.
- There is no way to constrain derivatives during interpolation. For example, it is possible for commanded velocities to exceed the capabilities of the actuator. Since the solution of the Hermite polynomial is unique, there is no way to "adjust" or re-calculate the trajectory with identical start and end conditions. What we are looking for in this case is a monotonic interpolation solution.
- The values on the spline are subject to oscillations if the time between x_0 and x_f is very small. For instance, it may be noted that in Figure 3(b), oscillations are amplified towards the very end, when noise levels are high. This is particularly pronounced due to the high order of the polynomial (5th). Although this can potentially be rectified by decreasing the order of the Hermite polynomial selectively, the minimum jerk requirement cannot be enforced in this case.

Several of the fore-mentioned issues are due to the fact that Hermite polynomials are direct interpolation functions, i.e. the interpolant function will directly pass through all knot or control points. Bézier Splines, that are 'parametric' curves provide a potential remedy to these problems by virtue of some of their unique properties: The start and end conditions can still be constrained by the position of two control points, but additional control points (that in general do not lie on the interpolant) not only specify the value of derivatives but also define an envelope, or convex hull. It is guaranteed that between the start and end point, the Bézier curve is always contained within that envelope. The calculation of the intermediate values is accomplished through De Casteljau's Algorithm, but is potentially more computationally expensive compared to the Hermite splines. Designing these Bézier curves for this problem and investigating the fore-mentioned claim form a part of the future and ongoing research directions of this work.

REFERENCES

Bauml, B., Wimbock, T., and Hirzinger, G. (2010). Kinematically Optimal Catching a Flying Ball with a Hand-Arm-System. In International Conference on Intelligent Robots and Systems (IROS), pages 2592–2599. IEEE.

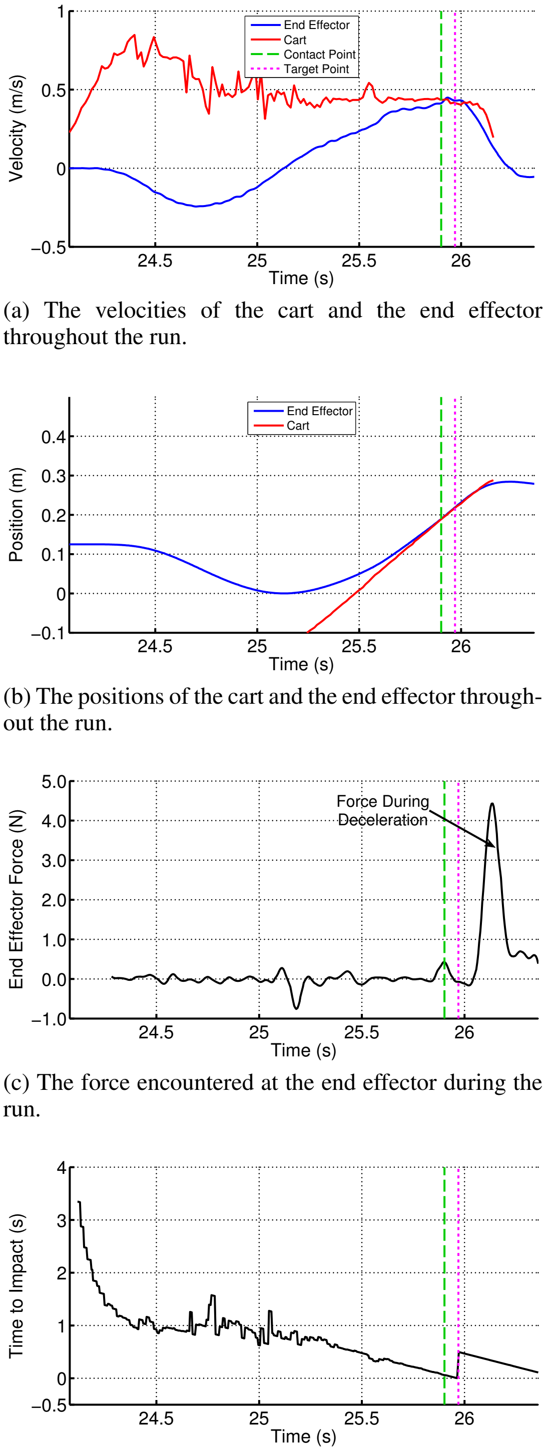
Borg, J., Mehrandezh, M., Fenton, R. G., and Benhabib, B. (2002). Navigation-Guidance-Based Robotic Interception of Moving Objects in Industrial Settings. Journal of Intelligent & Robotic Systems, 33(1):1–23.

Campos, J. A. F., Romo, S. R., Orozco, O. I., and Ortega, J. L. V. (2006). Robot Trajectory Planning for Multiple 2D Moving Objects Interception: A Functional Approach. In Electronics, Robotics and Automotive Mechanics Conference (CERMA), volume 1, pages 77–82. IEEE.

Croft, E. A., Fenton, R. G., and Benhabib, B. (1998). Optimal Rendezvous-Point Selection for Robotic Interception of Moving Objects. IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics, 28(2):192–204.

Gasparetto, A. and Zanotto, V. (2008). A Technique for Time-Jerk Optimal Planning of Robot Trajectories. Robotics and Computer-Integrated Manufacturing, 24(3):415–426.

Imai, Y., Namiki, A., Hashimoto, K., and Ishikawa, M. (2004). Dynamic Active Catching using a High-Speed Multifingered Hand and a High-Speed Vision Sys-



(a) The velocities of the cart and the end effector throughout the run.

(b) The positions of the cart and the end effector throughout the run.

(c) The force encountered at the end effector during the run.

(d) The changing estimates for the remaining time to impact.

Figure 9: Behavior of velocity matching controller when estimated cart velocity changes unpredictably. Please note that in all subfigures, the desired contact position (*Target Point*) and actual contact position (*Contact Point*) are marked by vertical lines.

tem. In International Conference on Robotics and Automation (ICRA), pages 1849–1854. IEEE.

Kövecses, J., Cleghorn, W. L., and Fenton, R. G. (1999). Dynamic Modeling and Analysis of a Robot Manipulator Intercepting and Capturing a Moving Object , with the Consideration of Structural Flexibility. Multibody System Dynamics, 3(2):137–162.

Lin, Z., Zeman, V., and Patel, R. (1989). On-line Robot Trajectory Planning for Catching a Moving Object. In International Conference on Robotics and Automation (ICRA), pages 1726–1731. IEEE Comput. Soc. Press.

Macfarlane, S. and Croft, E. A. (2003). Jerk-Bounded Manipulator Trajectory Planning: Design for Real-Time Applications. IEEE Transactions on Robotics and Automation, 19(1):42–52.

Nagendran, A., Crowther, W., and Richardson, R. C. (2011). Dynamic Capture of Free-Moving Objects. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 225(8):1054–1067.

Nagendran, A., Richardson, R., and Crowther, W. (2007). Bell Shaped Impedance Control to Minimize Jerk While Capturing Delicate Moving Objects. In Proceedings of ICINCO-RA, pages 504–511.

Namiki, A., Imai, Y., Ishikawa, M., and Kaneko, M. (2003). Development of a High-Speed Multifingered Hand System and its Application to Catching. In International Conference on Intelligent Robots and Systems (IROS), volume 3, pages 2666–2671. IEEE.

Nelson, B., Morrow, J., and Khosla, P. (1995). Fast Stable Contact Transitions with a Stiff Manipulator using Force and Vision Feedback. In International Conference on Intelligent Robots and Systems (IROS), pages 90–95. IEEE Comput. Soc. Press.

Papadopoulos, E., Tortopidis, I., and Nanos, K. (2005). Smooth Planning for Free-floating Space Robots Using Polynomials. In International Conference on Robotics and Automation (ICRA), pages 4272–4277. IEEE.

Piazzi, A. and Visioli, A. (1997). An Interval Algorithm for Minimum-Jerk Trajectory Planning of Robot Manipulators. In Conference on Decision and Control (CDC), volume 2, pages 1924–1927. IEEE.

Tang, Z., Zhou, C., and Sun, Z. (2003). Trajectory Planning for Smooth Transition of a Biped Robot. In International Conference on Robotics and Automation (ICRA), pages 2455–2460. IEEE.

Wang, C.-H. and Horng, J.-G. (1990). Constrained minimum-time path planning for robot manipulators via virtual knots of the cubic B-spline functions. IEEE Transactions on Automatic Control, 35(5):573–577.

Zhang, M. and Buehler, M. (1994). Sensor-based Online Trajectory Generation for Smoothly Grasping Moving Objects. In International Symposium on Intelligent Control, pages 141–146. IEEE.