2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Digital Hardware Implementation of an Active Disturbance Rejection Controller for a Highly Dynamic Parallel Orientation Manipulator

Taufiqur Rahman, Dion Hicks, M. Raju Hossain, and Nicholas Krouglicof

Abstract— **This paper details the development of a tracking controller for a highly dynamic parallel orientation manipulator that is capable of achieving high angular acceleration. The adopted control algorithm is derived from the active disturbance rejection control (ADRC) technology. The ADRC algorithm must be evaluated on the control hardware at a frequency consistent with the dynamics of the system without creating any inter-channel latency for a successful implementation. In this regard, the field programmable gate array (FPGA) is preferred because of its superior speed and parallelism. While there is a clear demand for FPGA controllers in the military and the aerospace domain because of the improved SWaP (size, weight, and power) capabilities, the literature provides a few examples of such controllers that mostly employ expensive high logic density FPGA chips. In contrast, this paper describes how an advanced controller can be efficiently prototyped on a relatively low-cost, low logic density FPGA hardware. The controller was tested in a co-simulation approach (wherein the digital implementation of the control algorithm and the dynamics of the manipulator are simulated together) to ensure a robust FPGA implementation. In addition, a hardware evaluation was also conducted using a linear voice coil actuator under varying inertial loads. Experimental results obtained from the simulation study and hardware testing confirm the robust performance of the designed controller.**

I. Introduction

Tracking control of a parallel orientation manipulator is generally difficult to achieve because of its nonlinear, time-varying, cross-coupled MIMO (multiple input multiple output) dynamics. However, several control schemes dealing with this particular problem have been presented in the literature. These controllers can be categorized into two classes; namely, the classical PID-type (error-driven) controllers (e.g., [1], [2]), and the modern controllers (e.g., [3]–[5]). An error-driven controller is generally not capable of delivering the robust performance required for negotiating the nonlinear dynamics of a strongly coupled MIMO system. In contrast, most modern controllers rely on detailed mathematical models of physical systems [6] to overcome this challenge. A successful implementation of these controllers requires the corresponding mathematical models to observe two competing constraints: (a) being able to provide sufficiently accurate estimation of the actual dynamics, and (b) being computationally simple enough to be evaluated in the control hardware at an adequate speed. Since the model of a parallel manipulator has to account for the multi-body dynamics resulting from a closed-loop kinematic structure, a corresponding implementation is computationally cumbersome, which is not conducive to real-time performance. As a result, model simplification becomes necessary to an extent that the simplified model may no longer represent the physical system accurately enough for the controller to be effective. In such a case, active disturbance rejection control (ADRC) offers an attractive solution that promises robust performance without the requirement of a system model.

In this paper, ADRC was applied for a miniature orientation manipulator that is capable of achieving high angular acceleration of the payload. Correspondingly, an effective controller must be fast enough to take full advantage of the dynamics of the system. In addition, latency among the input channels is undesirable, since it may compound the adverse effect of inter-channel coupling further. Ideally, a feasible control hardware platform must be able to execute the control algorithm at a sufficiently high speed in such a manner that all inputs to the system are dispatched simultaneously. Among all the available choices for control hardware, the field programmable gate array (FPGA) offers the speed and the parallelism required in this particular case. Moreover, an FPGA simplifies the prototyping process by integrating control circuitry for multiple input-output channels, ancillary communication modules, and soft processors for solving kinematics on a single chip. Thus, the FPGA is identified as the most suitable platform for implementing ADRC for a high speed manipulator. In addition to these advantages of an FPGA hardware, it generally offers superior reliability and robustness without the requirement of any additional cooling. In contrast, while a powerful multi-core processor may overcome the challenge of achieving parallelism by evaluating the control algorithm at a high speed utilizing its multiple CPU cores, it is considered unsuitable for embedded applications, especially in the military and the aerospace domain, because of reliability and robustness issues that originate from the software overhead (e.g., operating system) and the power requirements. However, developing a control system for an FPGA target is essentially an exercise in digital circuit design, which renders this task to be entirely different from developing computer codes to implement a preferred control algorithm. The literature provides a few examples where ADRC algorithm was implemented on an FPGA chip. The paper improves on the existing FPGA implementation of ADRC controllers by proposing an efficient design that does not require expensive high logic density FPGA chips.

Although mathematical correctness of an FPGA design can be verified in simulation under arbitrary test cases, the dynamics of the system, which is a function of the inputs, and the past states, is hardly random. Therefore, the controller must be verified and evaluated under dynamic conditions, which involves either the physical system, or a corresponding simulation model. Since hardware testing of

978-1-4799-3684-7/14/$31.00 ©2014 IEEE

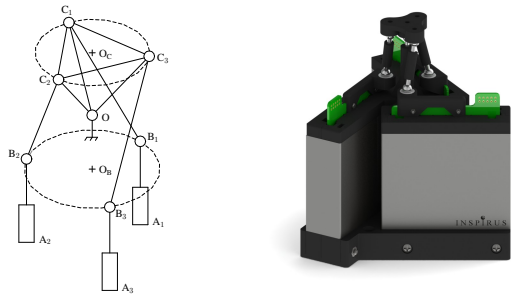
an unverified controller is difficult without access to all the state variables, this paper adopts a co-simulation approach wherein the manipulator model and the FPGA implementation of the controller are respectively simulated in a conventional simulator (e.g., MATLAB-Simulink) and an HDL (hardware description language) simulator (e.g., Modelsim). Both of the simulators communicate through the FLI (foreign language interface) feature of the HDL simulator for synchronization and data exchange. This approach enhances the visibility of the signals/quantities internal to the FPGA design, which is otherwise unobtainable from an FPGA and the real system. It should be noted that comprehensive access to these quantities is necessary for the sake of timing and mathematical verification, debugging, tuning, performance evaluation, etc. Moreover, HDL simulators can provide one to one correspondence in terms of input-output relations with the physical embodiment of the digital design on an FPGA chip provided that timing issues are completely resolved by the synthesizer (e.g., Quartus II). As a result, the controller running in an HDL simulator *exactly* represents the behavior of the FPGA hardware. However, the system model still suffers from uncertainties arising from unmodeled dynamics and inaccurate parameter estimations, which ADRC can handle [7]–[9].

The remainder of the paper is organized as follows. The construction of the system model is described in Section II. Mathematical formulation of the ADRC technology is discussed briefly in Section III. Section IV details the design of digital hardware for the controller. In addition, Section V provides a tuning procedure for the ADRC. The performance of the controller is experimentally verified and simulated in Section VI and the results are discussed in Section VII. Finally, the concluding remarks are offered in Section VIII.

II. System Model

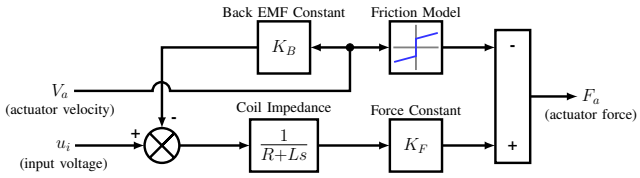
The parallel orientation manipulator in this paper features a 3P-S-S/S architecture as shown in Fig. 1. A dynamic model of the manipulator based on bond graph formalism was presented in [10]. A bond graph simulator that interfaces with an HDL simulator is not readily available. Alternatively, the multi-body model was reconstructed employing the SimMechanics toolbox of Simulink. Since one of the objectives of this paper is to describe how an advanced controller can be developed for an FPGA hardware, a detailed account on system modeling is considered to be outside of its scope and is not discussed further.

Three identical linear voice coil actuators (VCA) drive the orientation manipulator. Accordingly, the causal structure of the multi-body model was designed to accept actuation forces at the three prismatic joints. Since the multi-body model already accounts for the moving mass of each actuator, only the electrical and the frictional dynamics of the actuators are required to be incorporated with the multi-body manipulator model. Although the frictional characteristics of the spherical joints were ignored for the sake of simplicity, a friction model involving coulomb and viscous friction for each of the three actuated prismatic joints was implemented [see Fig. 2(a)],

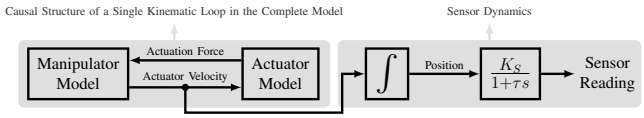


(a) Truss model (b) CAD model

Fig. 1. Kinematic structure of 3P-S-S/S orientation manipulator



(a) Electrical model of the voice coil actuator



(b) Model of a single kinematic loop & the sensor dynamics

Fig. 2. Block diagrams of the actuator model and the sensor model.

which is provided by [11],

$$F_f = \text{sign}(V_a)F_c + \mu_k V_a. \tag{1}$$

In (1), F_f, V_a, F_c, and μ_k denote total friction force, actuator velocity, coulomb friction force, and viscous friction coefficient respectively. The electrical dynamics of the actuators can be modeled as a RL (resistance-inductance) circuit. The corresponding transfer function (i.e., coil impedance) provides the coil current in terms of the coil voltage. As shown in Fig. 2(b), the actuator force and the actuator velocity respectively constitute the causal output and the causal input of the actuator sub-model.

An optical position sensor installed on each actuator provides the controller with the necessary feedback signals. The sensor acts as a first order filter with the transfer function $\frac{K_S}{1+\tau s}$, where K_S and τ denote the gain and the time constant of the sensor respectively. Because of the dynamics of the sensor, the feedback signal lags the actual position of the actuator. In order to verify the robustness of the controller against this feedback lag, the sensor dynamics was incorporated within the model [see Fig. 2(b)].

III. Active Disturbance Rejection Control

ADRC is a novel control paradigm proposed by Han [12] that packages the best features of classical and modern control theory in a single architecture. From classical control theory, it borrows the idea of error driven, rather than model-based, control law. In addition, it takes the best offering of modern

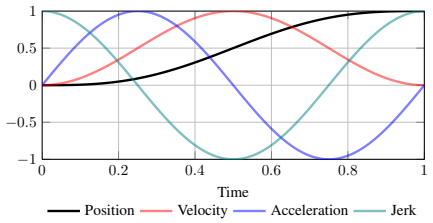


Fig. 3. Cycloidal motion profile for a unit displacement in unit time (all quantities have been scaled to have a maximum of unity except position).

control theory in the form of state observers. Thus, ADRC establishes itself to be a powerful control technology that is applicable to n-th order, nonlinear, time-varying, MIMO systems [6]. The success of ADRC can be attributed to its ability to estimate the plant dynamics and corresponding disturbances by an extended state observer (ESO), and to the subsequent active compensation for the estimated disturbances in the control effort [13].

An initial step in any controller prototyping exercise involves identification of a set of the system variables that are to be controlled (i.e., system output). In addition, a complementary set of system variables must be selected that are required to be manipulated in order to achieve the control objective (i.e., system input). In this case, the objective is to control the positions of the linear actuators by manipulating the corresponding actuator voltages so that the desired orientation of the payload can be achieved. Accordingly, the position sensor signals and the actuator input voltages respectively constitute the inputs and the outputs of the controller. Since each input to this MIMO system exhibits an obvious pairing with a unique output (i.e., input voltage and position of each actuator), the corresponding ADRC controller can be constructed by designing three individual SISO (single input single output) controllers for each input-output pair [14]. The position of the actuated joint in each of three kinematic loops is managed by a SISO controller.

The formulation of the control problem in this paper closely follows [12]. An empirical model of a representative kinematic loop of the parallel manipulator is a second order system of the following structure,

$$\ddot{y}_i = f_i\left(y_i, \dot{y}_i, \zeta_i, \dot{\zeta}_i, w_i(t), t\right) + b_i u_i, \tag{2}$$

where y_i denotes output of the i-th loop (i.e., actuator position), t denotes time, b_i is a system parameter, and u_i is the input to the i-th loop ($i = 1, 2, 3$). In addition, ζ_i denotes a vector of a comprehensive set of dynamic variables (excluding y_i) originating from the system model. $f_i(\cdot)$ represents the cross-coupled, nonlinear loop dynamics including external disturbance $w_i(t)$. It should be emphasized that $f_i(\cdot)$ does not decouple i-th loop dynamics from the other two kinematic loops. It is merely a lumped estimation of the aggregate effects of loop dynamics, cross-coupling, and external disturbance. It should also be noted that explicit knowledge of $f_i(\cdot)$ is not necessary for ADRC.

The system in (2) can be rewritten in state-space as in (3)

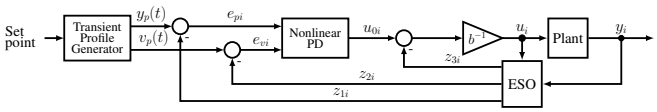


Fig. 4. Topology of ADRC for a single kinematic loop.

with an augmented state $x_{3i} = f_i(\cdot)$.

$$\begin{cases} \dot{x}_{1i} = x_{2i} \\ \dot{x}_{2i} = x_{3i} + b_i u_i, \quad x_{3i} = f_i(\cdot) \\ \dot{x}_{3i} = \dot{f}_i(\cdot) \\ y_i = x_{1i} \end{cases} \tag{3}$$

In the context of feedback control, the *total disturbance* $f_i(\cdot)$ needs to be overcome by the control effort u_i in order to achieve desired behavior of y_i [12] . To this end, an extended state observer (ESO) for the system in (3) can be constructed in the following form.

$$\begin{cases} e_i = z_{1i} - y_i \\ \dot{z}_{1i} = z_{2i} - \beta_{01} e_i \\ \dot{z}_{2i} = z_{3i} + b_i u_i - \beta_{02}\vartheta_{1i}(e_i) \\ \dot{z}_{3i} = -\beta_{03}\vartheta_{2i}(e_i) \end{cases} \tag{4}$$

Here, z_{1i}, z_{2i}, and z_{3i} are respectively the observations of x_{1i}, x_{2i}, and x_{3i}. In addition, β_{01}, β_{02}, and β_{03} are the tuning parameters for the ESO. $\vartheta_{1i}(e_i)$ and $\vartheta_{2i}(e_i)$ are appropriate linear or nonlinear functions of the tracking error e_i. In this case, they were chosen as $\vartheta_{1i}(e_i) = \vartheta_{2i}(e_i) = e_i$ in order to linearize the ESO. The observation of $f_i(\cdot)$ provided by the ESO allows the control law $u_i = \frac{u_{0i} - f_i(\cdot)}{b_i} \approx \frac{u_{0i} - z_{3i}(\cdot)}{b_i}$ to be applied on the system in (3), which ultimately results in a reduced system of cascade integral form as in (5).

$$\dot{x}_{1i} = x_{2i}, \quad \dot{x}_{2i} = u_{0i}, \quad y_i = x_{1i} \tag{5}$$

This reduced system can be controlled by calculating u_{0i} as a function of the tracking error and its derivate (i.e., a PD controller). In addition to the central idea of estimating and rejecting disturbances, ADRC topology also features a few other components that are discussed below.

A. Transient Profile

Each SISO loop of the controller in this paper is augmented with a transient profile generator producing a trajectory that the actuators can follow without exceeding their actuation capacities. Although a number of methods for generating transient profiles is available in the literature, only the limited jerk (derivative of acceleration) profiles are of interest in the context of the parallel manipulator. The limited jerk transient motion profile improves tracking performance, and reduces mechanical impact [15]. It should be noted that this paper does not implement trajectory planning in the workspace coordinates because the classic ADRC formulation involves an empirical model of system dynamics [i.e., see (2)] in terms of the input (e.g., actuator voltage) and the measured output of the plant (e.g., actuator position). Correspondingly, trajectory planning in the joint space coordinates is considered to be

more conducive to the structure of the ADRC controller and is the approach adopted in this paper.

A transient trajectory generating limited jerk can be modeled by a sinusoidal acceleration profile that produces a cycloidal displacement [16] as shown in Fig. 3. The displacement, and the corresponding velocity of the cycloidal profile are provided as a function of time in (6) and (7).

$$y_p(t) = (y_f - y_j)\left(\frac{t}{t_f} - \frac{1}{2\pi}\sin 2\pi\frac{t}{t_f}\right) + y_j \qquad (6)$$

$$v_p(t) = \frac{y_f - y_j}{t_f}\left(1 - \cos 2\pi\frac{t}{t_f}\right) \qquad (7)$$

Here, y_f is the final position (i.e., desired set point), y_j is the initial position, t is the current time ($t = 0$ at the onset of motion), and t_f is the *transient time* (i.e., at time $t = t_f$ the position trajectory reaches y_f).

B. Nonlinear PD Controller

ADRC employs a nonlinear weighted sum (n-PD) of the position and velocity tracking errors in order to calculate u_{0i} as follows,

$$u_{0i} = k_p fal(e_{pi}, \alpha_p, \delta_p) + k_d fal(e_{vi}, \alpha_v, \delta_v). \qquad (8)$$

Here, e_{pi} and e_{vi} are respectively the position and the velocity tracking error for the i-th kinematic loop. $fal(\cdot)$ is a nonlinear function provided by (9), and parameterized with α and δ.

$$fal(e, \alpha, \delta) = \begin{cases} \frac{e}{\delta^{1-\alpha}}, & |e| \le \delta \\ |e|^{\alpha}\,\mathrm{sign}\,(e), & |e| > \delta \end{cases}, \text{ with } \delta > 0 \qquad (9)$$

Gao *et al.* in [17] provides a detailed explanation on how the $fal(\cdot)$ function improves controller performance by guarding against control signal saturation. It also minimizes steady state error without integral control. The structure of one of the three SISO loops of the controller is provided in Fig. 4.

IV. FPGA Prototyping of ADRC

Although ADRC has been adopted for parallel manipulators (e.g., [9]), a corresponding FPGA implementation is rare. However, ADRC systems intended for other applications are reported to employ FPGA hardware (e.g., [13], [18]). This paper differs from prior art in many aspects, principally in the application of fixed point arithmetic to evaluate the necessary transcendental functions in a piece-wise linear manner. Since this design philosophy encourages efficient utilization of FPGA resources, it is possible to implement the design on a low-cost, low logic density FPGA. It should be noted that floating point format generally provides good accuracy and dynamic range; however, comparable performance is also obtainable from fixed point format, if implemented carefully [19]. In this paper, VHDL was chosen as the preferred language. The design also utilizes the IEEE fixed point package extensively.

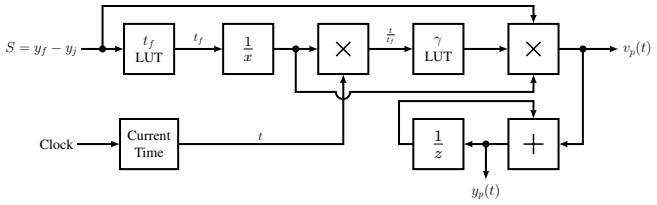


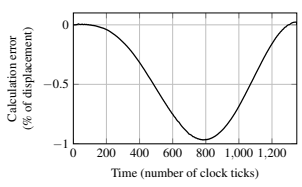
Fig. 5. Cycloidal motion profile generation on an FPGA.

A. Transient Profile Generator

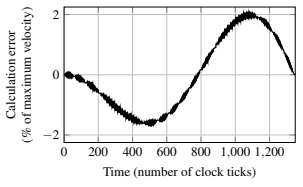
The transient time t_f in the cycloidal motion profile equations (6) and (7) determines how fast the desired position is reached. While a smaller t_f is preferred, the resulting jerk may be too great for the limited actuation capacity. Determining the optimum t_f for a given maneuver (i.e., displacement in at least one actuator) constitutes an interesting problem that is beyond the scope of the paper. However, this paper employs a LUT (look up table) based approach in order to determine t_f for a given maneuver. To this end, the stroke of each actuator was arbitrarily divided into four equal segments and different t_f values were assigned for each segment in order to construct the LUT. It should be noted that the t_f determined by the LUT for the largest of the three actuator displacements provides the global transient time for an arbitrary maneuver. Consequently, all three resulting motion profiles become synchronous in order to achieve the idea of *synchronous control* [2] without any extra effort.

Observing that (6) can be obtained by integrating (7), only the velocity profile needs to be implemented for an efficient design. The nonlinear component $\gamma = (1 - \cos 2\pi\frac{t}{t_f})$ in (7) poses the greatest challenge, since FPGA architecture is not conducive to operations involving transcendental functions. The bounded nature of its argument (i.e., $0 \le \frac{t}{t_f} \le 1$), however, permits implementation of the nonlinear component γ as a LUT. Since γ is symmetric about $\frac{t}{t_f} = \frac{1}{2}$, the LUT is constructed efficiently by evaluating γ for a set of values of $\frac{t}{t_f}$ uniformly distributed over the range $\left[0, \frac{1}{2}\right]$. Flexibility and scalability can be achieved by inferring this LUT analytically at compile time employing the IEEE real math package. A simplified block diagram of the corresponding implementation is provided in Fig. 5. In addition, Fig. 6 shows that the accuracy of the motion profile calculated in fixed point is adequate. Furthermore, the following remarks are deemed relevant:

a. Since the ADRC algorithm is evaluated once in each control clock cycle, it is natural to express the transient time t_f in terms of number of clock cycles. This allows the calculation of $\frac{t}{t_f}$ as a clock triggered accumulation of t_f^{-1}. It is also possible to store pre-calculated t_f^{-1} values in order to save the resources required for the inversion circuit.
b. Wherever possible, dedicated silicon resources should be employed for implementing the design. This includes on-chip memory and multiplier blocks.
c. Since the position sensor provides a 16-bit integer rep-

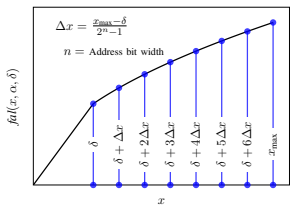


(a) Error of the position profile expressed as a percentage of the displacement.

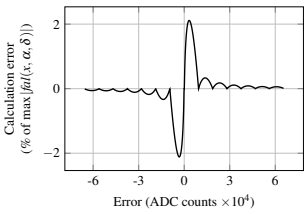


(b) Error of the velocity profile expressed as a percentage of the maximum velocity.

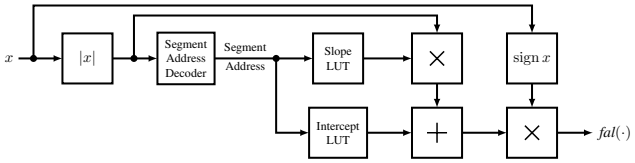
Fig. 6. Accuracy of the motion profile as calculated by the profile generator module with respect to a double precision floating point calculation. The displacement corresponds to the entire range of a 16 bit position sensor.



(a) Approximating $fal(x, \alpha, \delta)$ as a piece-wise linear function.



(b) Calculation error of $fal(x, \alpha, \delta)$ with respect to a double precision evaluation ($\alpha = 0.75, \delta = 100$).



(c) Evaluation of $fal(\cdot)$ on an FPGA.

Fig. 7. FPGA implementation and accuracy analysis of $fal(x, \alpha, \delta)$.

resenting the position (i.e., ADC counts), the velocity profile is calculated in terms of ADC counts per clock period instead of conventional units (e.g., mm/s, m/s). Subsequently, the position profile can be obtained by a simple clock triggered accumulation of the corresponding velocity profile.

B. Extended State Observer

Similar to [18], a linear ESO as in (10) is adopted for the controller.

$$\begin{aligned} e_i(k) &= z_{1i}(k) - y_i(k) \\ \sigma(k) &= T\left(z_{3i}(k) + b_i u_i(k)\right) \\ z_{1i}(k+1) &= z_{1i}(k) + T z_{2i}(k) - \beta_{01} e_i(k) \\ z_{2i}(k+1) &= z_{2i}(k) + \sigma(k) - \beta_{02} e_i(k) \\ z_{3i}(k+1) &= z_{3i}(k) - \beta_{03} e_i(k) \end{aligned} \tag{10}$$

Here, T is the sampling period of the discrete ESO. Since (10) involves only multiplication and addition operations, implementing it on an FPGA is straightforward.

C. Nonlinear Control Law

The nonlinearity in the control law originates from the $fal(x, \alpha, \delta)$ function in (9), which is nonlinear except when $|x| \leq \delta$. This observation inspires the idea of approximating it as a piece-wise linear function, where each linear piece is characterized by a slope and an intercept. Since $fal(\cdot)$ is an odd function, it is sufficient to characterize it for only positive valued arguments. To this end, a reasonable $x_{\max}$ is chosen arbitrarily and the range $[0, x_{\max}]$ is subsequently divided into 2^n segments, as shown in Fig. 7(a). Here, n corresponds to the address bus width of the look up tables that store the linear characteristics of each segment. For simplicity, the slope and the intercept characterizing each segment are evaluated for their respective terminal points only. The block diagram representation of the corresponding digital circuit is presented in Fig. 7(c). In addition, Fig. 7(b) shows that the adopted piece-wise linearization approach provides acceptable accuracy.

V. Tuning of ADRC

A simple tuning method for second order ADRC in [8] determines the n-PD, and the ESO gains as a function of the controller bandwidth ω_c, and the observer bandwidth ω_o respectively [20]. In addition, the tuning procedure of the system parameter b_i in (10) is also provided in [8]. Since a large controller bandwidth ω_c demands high dynamics from the system, the response tracks the set-point faster. However, this may lead to undesirable oscillations or instability because the system may be marginally capable or even completely incapable of delivering the demanded dynamics. Hence, the tuning exercise of ω_c attempts to find an acceptable compromise between requirements of performance and stability margin [21]. Similarly, a large observer bandwidth ω_o improves tracking performance of the ESO, which is achieved at the cost of a degraded noise tolerance. Thus, ω_o is tuned to balance tracking performance against the noise sensitivity of ESO [13].

According to [20], the controller gains in (8) are determined as,

$$k_p = \omega_c^2, \quad k_d = 2\omega_c. \tag{11}$$

It should be noted that the time scale in the velocity tracking error e_{vi} in (8) must be consistent with the time scale of ω_c. Since e_{vi} is expressed in ADC counts per clock period of the controller, the derivative gain in this paper is also rescaled accordingly as,

$$k_d = \frac{2\omega_c}{T}. \tag{12}$$

The observer gains $[\beta_{01}, \beta_{02}, \beta_{03}]$ for a discrete ESO are calculated in [22] as follows,

$$\begin{aligned} \beta_{01} &= 1 - \beta^3, \quad \beta = e^{-\omega_o T} \\ \beta_{02} &= (1-\beta)^2(1+\beta)\frac{3}{2T} \\ \beta_{03} &= (1-\beta)^3\frac{1}{T^2}. \end{aligned} \tag{13}$$

VI. Performance Evaluation of the ADRC Controller

The performance of the ADRC controller was not benchmarked against the widely used PID technology, since it was observed in [10] that acceptable performance from a

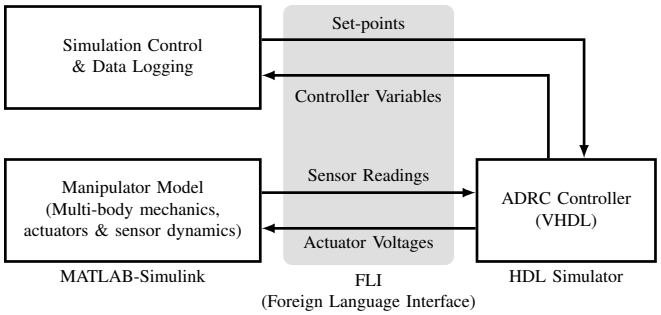
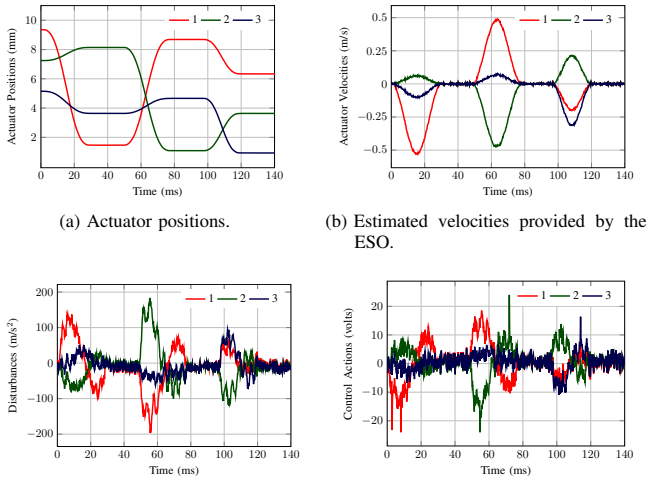


Fig. 8. Architecture of the co-simulation experiment.



(a) Actuator positions.

(b) Estimated velocities provided by the ESO.

(c) Estimated disturbances provided by the ESO.

(d) Disturbance corrected control actions.

Fig. 9. Simulated system responses in three consecutive maneuvers of the manipulator.

PID controller is difficult to achieve. However, the FPGA implementation of the controller was evaluated in a simulation experiment. The manipulator and the actuator models were complemented with appropriate system parameters that were either estimated from corresponding CAD models (e.g., inertia and geometric properties of the manipulator links), or measured from prototypes of different system components (e.g., actuator parameters). The parameters defining the simulation model are provided in Tables II and I. It should be noted that the manipulator geometry was derived in accordance with [23] assuming each actuator provides 12 mm of stroke [24].

In the simulation experiment, the designed ADRC controller was evaluated by an HDL simulator and the manipulator model in Simulink provided the ADRC controller with the simulated position data of all three actuators to calculate appropriate control actions. These control actions were then fed back to the Simulink model (i.e., co-simulation approach, refer to Fig. 8) to complete the control loop. The tuning procedure in Section V was followed in order to obtain the controller parameters provided in Table III. Subsequently, a Monte-Carlo experiment comprised of two hundred arbitrary maneuvers was conducted for control performance evaluation.

TABLE I

ACTUATOR PARAMETERS

	Parameter	Value	Unit
Experimentally Obtained	Motor Constant	3.31	N/amp
	Coil resistance	8.7	Ω
	Coil inductance	327E-06	H
	Sensor gain	5.43	ADC counts/μm
Assumed for Simulation	Coulomb friction force	0.5	N
	Viscous friction co-efficient	0.16	Ns/m
	Sensor gain	5.46	ADC counts/μm
	Sensor time constant	300	μs

TABLE II

PARAMETERS FOR THE MULTI-BODY DYNAMIC MODEL OF THE MANIPULATOR

Link	Mass	Principal Moments of Inertia
	(kg)	(Kg·m^2)
Moving Platform[a]	58.46E-03	[4.35E-06 4.64E-06 4.64E-06]
Intermediate Link	3.94E-03	[1.09E-08 7.58E-07 7.58E-07]
Actuator Moving Mass	31.96E-03	-

[a] with a payload of 50 gm

Settling times and absolute steady state errors of these maneuvers quantified the performance of the controller. The largest of the three individual settling times provided by each actuator determines the corresponding settling time t_{sm} for a given maneuver. The settling time is provided by the time elapsed from the initiation of the desired displacement to the time when the system response has entered and remained within ± 30 μm of the steady state value. In order to detect steady state in the time-series of the system response, the method of moving variance [25] was employed. When the moving variance of a window consisting of 200 samples was observed to be less than the variance threshold of 200 ADC counts (corresponds to a range of ± 5.2 μm with a 95% C.I.), the system was considered to have entered steady state. The corresponding steady state value was obtained by taking the mean of all responses sampled from the time when the system entered steady state to the time when the next maneuver is initiated. Based on the steady state values thus determined, the absolute steady state errors in a given maneuver for all three actuators $[e_{ss1}, e_{ss2}, e_{ss3}]$ were calculated. Statistics of the simulated performance metrics is provided in Table IV. Please note that a uniform random noise with a range of ± 8 ADC counts was introduced in each input channel of the controller to simulate the occurrence of noise in a practical sensor. Since the effect of inter-channel coupling is the strongest when one of the three actuators attempts to move in a direction opposite to the other two, the simulated system responses for three such cases are provided in Fig. 9.

Although hardware validation of the controller is necessary for a comprehensive evaluation, an experimental setup consisting of a prototype of the manipulator is currently under

TABLE III

TUNING PARAMETERS FOR THE CONTROLLER

Parameter	Value		Unit
	Simulation	Experiment	
ω_c	450	525	rad/s
ω_o	6500	1700	rad/s
b_i	0.75E+08	0.75E+08	-
T	48.0	48.0	μs
α_p	0.50	0.75	-
δ_p	50.0	30.0	ADC counts
α_v	2.00	1.25	-
δ_p	5.0	5.0	ADC counts/T

TABLE IV

PERFORMANCE OF THE CONTROLLER IN THE CO-SIMULATION STUDY

Statistic	Settling Time (ms)	Steady State Error (μm)		
	t_{sm}	e_{ss1}	e_{ss2}	e_{ss3}
Mean	24.72	2.7876	3.1355	2.8405
Max	33.70	39.5874	64.8370	66.3772
Min	16.22	0.0091	0.0430	0.0013
σ	04.69	4.5693	6.6044	5.7871

development. Hence, the controller was implemented on a DE0-Nano FPGA development board for a single actuator with a variable load. A host computer was interfaced with the FPGA through a serial link for experiment control and data acquisition. After a new position of the actuator had been initiated, the FPGA logged interesting controller variables for the next 98.26 ms (2047 ticks of the controller clock). The logged data was then sent to the host computer for storage and further analysis. Although this experiment cannot ascertain realistic controller performance for the manipulator, its robustness against varying inertia can still be demonstrated. To this end, the controller was tuned for a nominal plant with a moving mass of 24.55 gm and a Monte-Carlo experiment consisting of 200 trials was conducted. Without retuning the controller parameters, the experiment was repeated for a moving mass of 52.45 gm. Finally, the experimental data was analyzed to determine the performance metrics presented in Table V. Similar to the simulation experiment, the settling time t_{ss} and the steady state error e_{ss} in each trial were employed to quantify the control performance for that trial. The system responses featuring the settling times closest to their respective mean values are presented in Fig. 10 and Fig. 11.

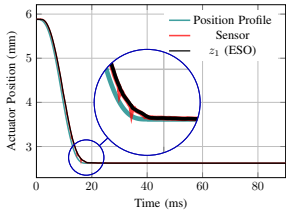
TABLE V

PERFORMANCE OF THE CONTROLLER IN HARDWARE EVALUATION

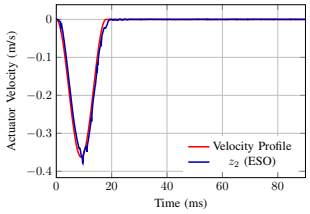
Statistic	Nominal Plant		Perturbed Plant	
	t_{ss} (ms)	e_{ss} (μm)	t_{ss} (ms)	e_{ss} (μm)
Mean	16.8950	3.0574	18.1493	5.2391
Max	27.1680	14.0397	42.8640	20.8269
Min	11.3760	0.0187	11.7120	0.0109
σ	3.4936	2.8496	4.7572	4.2609

VII. DISCUSSION

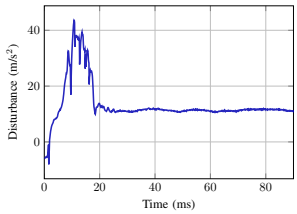
Since the validation of a complex digital circuit is not a trivial exercise, only a subset of all possible test cases is considered [26]. Because the number of possible test cases exponentially increases with the complexity of the circuit, an exhaustive validation is often impractical. Nonetheless, the co-simulation study validates the controller in terms of functionality and structural accuracy of the digital design under simulated operating conditions. As a result, the risk of damaging an experimental setup with a controller that would have been otherwise validated for only random inputs is greatly minimized. It also offers better debugging capacity by



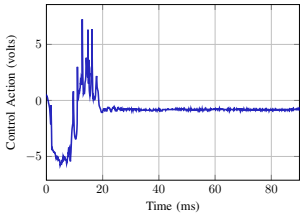
(a) Position (inset: impulse noise in the sensor signal is successfully filtered by the ESO).



(b) Estimated velocity provided by the ESO.

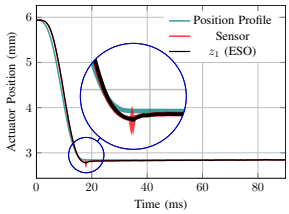


(c) Estimated disturbances provided by the ESO.

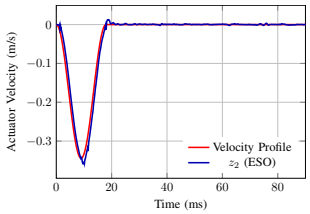


(d) Disturbance corrected control action.

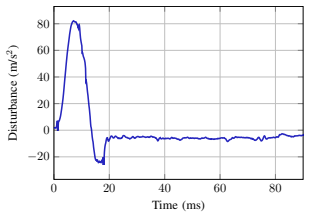
Fig. 10. Recorded system responses in a representative experimental trial of the nominal plant.



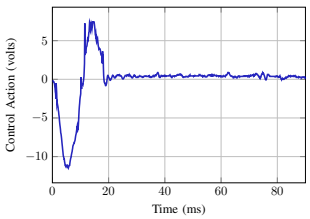
(a) Position (inset: impulse noise in the sensor signal is successfully filtered by the ESO).



(b) Estimated velocity provided by the ESO.



(c) Estimated disturbances provided by the ESO.



(d) Disturbance corrected control action.

Fig. 11. Recorded system responses in a representative experimental trial of the perturbed plant.

providing comprehensive access to quantities internal to the controller. Thus, a co-simulation study appreciably streamlines the prototyping process of a complex controller on digital hardware by reducing development time and resources.

Admittedly the data presented in Tables IV and V are not absolute measures of control performance, since these performance metrics depend on the steady state detection algorithm. However, it was confirmed through simulation that active disturbance rejection control can perform well for a nonlinear, time-varying, highly coupled dynamic system. In addition, the data presented in Table V suggest that the controller delivers similar performance for both the nominal and the perturbed plant despite an increase in the moving mass by more than 100% in the perturbed plant. This observation is indicative of the robustness in control performance, which is a prerequisite for handling the time-varying inertia of the orientation manipulator.

In the experimental hardware, impulse noise was observed in the sensor data [see Fig. 10(a) and Fig. 11(a)]. As suggested in [20], a moderate transient profile was adopted for the hardware experiment in order to mitigate the adverse effect of noise on control performance.

VIII. Conclusion

This paper details the digital design of an ADRC controller that involves transcendental functions employing the efficient fixed point format. Besides demonstrating how the co-simulation approach can simplify the development flow of digital systems that are designed to be coupled with dynamic systems, this paper experimentally establishes the robustness of the controller in terms of managing varying inertia. However, the controller performance must still be verified on hardware for a prototype of the orientation manipulator. Correspondingly, future work involves constructing a prototype of the manipulator, and subsequently evaluating the practical control performance. In addition, further optimization of the controller will also be explored.

References

[1] T. Villgrattner, E. Schneider, P. Andersch, and H. Ulbrich, "Compact high dynamic 3 dof camera orientation system: Development and control," *Journal of System Design and Dynamics*, vol. 5, pp. 819–828, 2011.

[2] Y. Su, D. Sun, L. Ren, and J. Mills, "Integration of saturated PI synchronous control and PD feedback for control of parallel manipulators," *Robotics, IEEE Transactions on*, vol. 22, no. 1, pp. 202 – 207, feb. 2006.

[3] A. Vivas and P. Poignet, "Model based predictive control of a fully parallel robot," in *7th IFAC Symposium on Robot Control (SYROCO03)*, Semptember 2003, pp. 277–282.

[4] J. Dong, S. Salapaka, and P. Ferreira, "Robust MIMO control of a parallel kinematics nano-positioner for high resolution high bandwidth tracking and repetitive tasks," in *Decision and Control, 2007 46th IEEE Conference on*, dec. 2007, pp. 4495 –4500.

[5] O. Linda and M. Manic, "Uncertainty-robust design of interval type-2 fuzzy logic controller for delta parallel robot," *Industrial Informatics, IEEE Transactions on*, vol. 7, no. 4, pp. 661 –670, nov. 2011.

[6] Z. Gao, "Active disturbance rejection control: a paradigm shift in feedback control system design," in *American Control Conference*, June 2006, p. 7.

[7] A. Martini, F. Leonard, and G. Abba, "Robust nonlinear control and stability analysis of a 7DOF model-scale helicopter under vertical wind gust," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, Sept. 2008, pp. 354 –359.

[8] C. Xing, L. Donghai, G. Zhiqiang, and W. Chuanfeng, "Tuning method for second-order active disturbance rejection control," in *Control Conference (CCC), 2011 30th Chinese*, July 2011, pp. 6322 –6327.

[9] Y. X. Su, B. Y. Duan, C. H. Zheng, Y. F. Zhang, G. D. Chen, and J. W. Mi, "Disturbance rejection high-precision motion control of a stewart platform," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 3, pp. 364–374, May 2004.

[10] T. Rahman, G. Rideout, and N. Krouglicof, "Evaluation of dynamic performance of nonspherical parallel orientation manipulators through bond graph multi-body simulation," in *10th International Conference on Bond Graph Modeling and Simulation (ICBGM'2012)*, July 2012.

[11] S. T. Karris, *Introduction to Simulink with Engineering Applications*. Orchard Publication, 2011, ch. Appendix E, pp. E–18.

[12] J. Han, "From PID to active disturbance rejection control," *Industrial Electronics, IEEE Transactions on*, vol. 56, no. 3, pp. 900 –906, March 2009.

[13] Q. Zheng, L. Dong, D. H. Lee, and Z. Gao, "Active disturbance rejection control for MEMS gyroscopes," *Control Systems Technology, IEEE Transactions on*, vol. 17, no. 6, pp. 1432 –1438, Nov. 2009.

[14] Z. Gao, "Controllers, observers, and applications thereof," US Patent US 8 060 340 B2, Nov, 2011.

[15] S. Macfarlane and E. Croft, "Jerk-bounded manipulator trajectory planning: design for real-time applications," *Robotics and Automation, IEEE Transactions on*, vol. 19, no. 1, pp. 42 – 52, feb 2003.

[16] R. L. Norton, *CAM Design and Manufacturing Handbook*. Industrial Press Inc., 2009.

[17] Z. Gao, Y. Huang, and J. Han, "An alternative paradigm for control system design," in *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, vol. 5, 2001, pp. 4578 –4585 vol.5.

[18] Z. Ping and Z. Gao, "An FPGA-based digitial control and communication module for space power mangement and distributed systems," in *2005 Americal Control Conference*, 2005.

[19] F. de Dinechin, J. Detrey, O. Cret, and R. Tudoran, "When fpgas are better at floating-point than microprocessors," in *Proceedings of the 16th international ACM/SIGDA symposium on Field programmable gate arrays*, ser. FPGA '08. New York, NY, USA: ACM, 2008, pp. 260–260. [Online]. Available: http://doi.acm.org/10.1145/1344671.1344717

[20] Z. Gao, "Scaling and bandwidth-parameterization based controller tuning," in *American Control Conference, 2003. Proceedings of the 2003*, vol. 6, June 2003, pp. 4989 – 4996.

[21] Z. Qing and G. Zhiqiang, "On practical applications of active disturbance rejection control," in *Control Conference (CCC), 2010 29th Chinese*, july 2010, pp. 6095 –6100.

[22] R. Miklosovic, A. Radke, and Z. Gao, "Discrete implementation and generalization of the extended state observer," in *Proceedings of the 2006 Americal Control Conference*, Minneapolis, Minnesota, June 2006.

[23] T. Rahman, N. Krouglicof, and L. Lye, "Kinematic synthesis of nonspherical orientation manipulators: Maximization of dexterous regular workspace by multiple response optimization," *Journal of Mechanical Design*, vol. 134, no. 7, p. 071009, 2012. [Online]. Available: http://link.aip.org/link/?JMD/134/071009/1

[24] N. Krouglicof, M. Morgan, N. Pansare, T. Rahman, and D. Hicks, "Development of a novel pcb-based voice coil actuator for opto-mechatronic applications," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, Nov 2013, pp. 5834–5840.

[25] M. Kim, S. H. Yoon, P. A. Domanski, and W. V. Payne, "Design of a steady-state detector for fault detection and diagnosis of a residential air conditioner," *International Journal of Refrigeration*, vol. 31, no. 5, pp. 790 – 799, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140700707002186

[26] R. Ho and M. Horowitz, "Validation coverage analysis for complex digital designs," in *Computer-Aided Design, 1996. ICCAD-96. Digest of Technical Papers., 1996 IEEE/ACM International Conference on*, 1996, pp. 146–151.