MECHATRONIC MODEL-BASED DESIGN APPLIED TO AN H-BOT ROBOT

Kevin Craig

Department of Engineering, Hofstra University, Hempstead, NY, USA

ABSTRACT

Mechatronic system design is multidisciplinary and integration among the mechanical, sensor, actuator, electronic, computer, and control elements is essential. The integration is done simultaneously from the start of the design process and the design is model-based. Modeling, physical and mathematical, is the key in modern engineering practice. Accurate motion control and energy efficiency in industrial machines heavily depend on trajectory planning and the appropriate selection of the motors controlling the axes of the machine. A model-based design approach is proposed for (1) trajectory planning that leads to accurate positioning and energy efficiency and (2) optimized selection of motors prior to building a prototype. As planar positioning is an important task in industrial applications, a two-axis, single-belt-driven, H-frame planar positioning robot called an H-Bot was built. The proposed approach is demonstrated via modeling, analysis, control-design simulation using MatLab / Simulink, and hardware implementation using the Arduino and LabVIEW MyRIO.

1. INTRODUCTION

Mechatronics is the best practice by engineers driven by the needs of industry and human beings. It is technology integration to achieve optimal system functionality, the synergistic integration of physical systems, electronics, controls, and computers through the design process, from the very start of the design process, thus enabling complex decision making. Integration is the key element in mechatronic design as complexity has been transferred from the mechanical domain to the electronic and computer software domains. Mechatronics is an evolutionary design development that demands horizontal integration among the various engineering disciplines, as well as vertical integration between design and manufacturing, and is what modern mechanical engineering needs to be [11].

As a professor at Rensselaer Polytechnic Institute from 1989-2008, I created the mechatronics undergraduate and graduate programs there, and, in 1995, created the Mechatronics diagram (Figure 1) now used around the world to illustrate this multidisciplinary field. Figure 2 shows the model-based Mechatronic System Design Process I created [2]. In model-based design, computer simulation without experimental verification is, at best, questionable, and, at worst, useless [3].
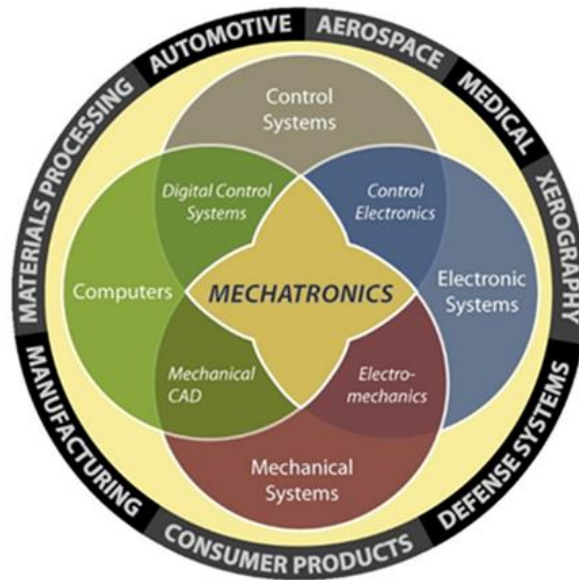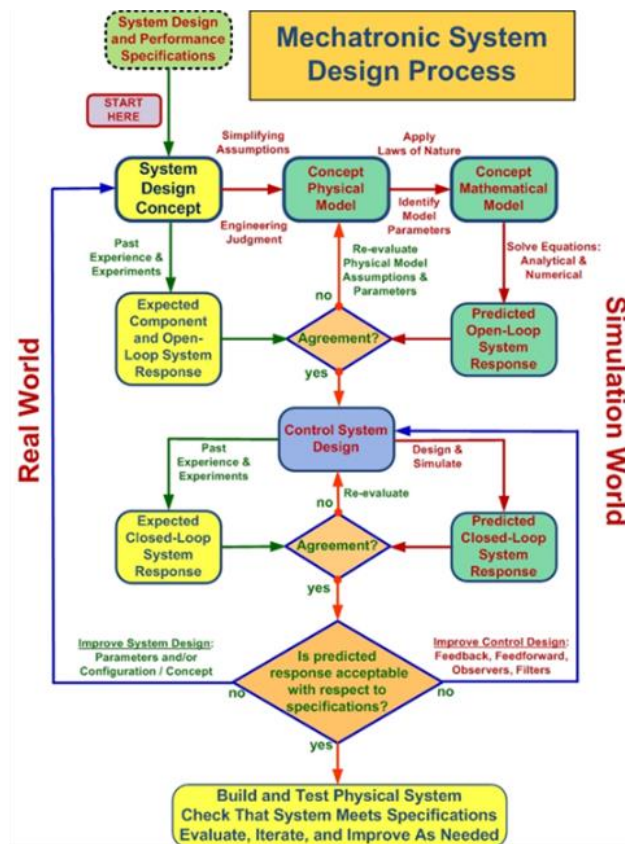
Figure. 1 Mechatronics Diagram

Figure 2. Mechatronic System Design Process

Model-Based Mechatronic System Design (Figure 2) re-engineers the traditional development process from one which is paper-based to one that uses an executable model that is the repository for all information about the concept, design, and implementation. The model is used throughout

the four stages of development: research, design, implementation, and verification and validation. At each stage of development, the model is updated and elaborated ensuring continuity and traceability throughout the evolution of the design. In its most basic form, model-based system design is the use of models to describe the specifications, operation, and performance of a component or system of components.

Characteristics of model-based system design include: the capability to inexpensively design and test multiple approaches without costly commitment to prototype hardware early in the development process; a collaborative design environment using common executable specifications that connect to requirement documents and lets all multiple engineering disciplines communicate in a common language; the ability to reduce development costs by easily finding and correcting errors during an early simulation stage; and the capability to develop complex embedded systems that provide customer value, product quality, and sophistication in multidisciplinary systems.

Model-based system design has been of fundamental importance for industrial development during the past few decades as the needs for energy efficiency, reliability, flexibility, and accuracy have increased to satisfy a market that is demanding higher productivity at reduced costs in a sustainable manner [10]. Tighter, more complex, and challenging specifications need to be achieved. As machines become more complex, model-based design has helped engineers overcome the challenges in mechatronic system design. In a mechatronic system design approach, many stages of the design process overlap and all disciplines including mechanical, electrical, control and computer systems are fully integrated to enhance the synergy among them. This is leveraged by virtual prototyping and modeling that allows the integration, evaluation, and simulation of various scenarios before a system is built.

A critical issue that machine builders face in the design stage is the selection of motion profiles, motors, transmission systems, and electronics to control each axis of a machine. To overcome this challenge, a model-based design approach to properly select motion profiles and motors for a machine prior to building a prototype is presented in this paper [1, 7, 8]. In this approach, the inverse kinematics of the system, in conjunction with trajectory planning, is used to compute the motion profiles of the motors. These motion profiles feed the inverse kinetic model of the system that is simulated in open loop to compute the required motor torque vs. speed curves. These are used to select candidate motors. After selecting candidate motors, the inertia and losses of the motor and transmission system are included with the control design in a closed-loop simulation to validate the solution and stability of the system. This approach allows simulating not only the mechanical system, but also the control scheme of the system in an integrated manner. This approach can also help uncover issues before building the prototype, in addition to closing the gap between the mechanical and control designs.

The proposed motion profile and motor selection process is demonstrated in this paper by selecting the motion profile and motors for a planar robot called an H-Bot, which consists of two motors, a timing belt, and rails mounted to form the shape of an H [5, 6, 14, 15]. The physical system of this robot is shown in Figure 3. An H-bot is a two-dimensional robot extensively used in industry in applications such as pick-and-place, sorting, gluing, and inspection. This type of robot is particularly attractive for machine builders due to the relatively ease of manufacturing. Although this motion profile and motor selection process will be demonstrated with a robot, it is a generic method that can be applied to any industrial machine.

An appropriate motion profile is not only critical for the motor selection process, but also for the overall performance of the system. However, motion systems often use jerky motion profiles that

stress the machine and motors, and produce unwanted vibrations and wear, which result in poor performance and shortened life. This is due to the use of inappropriate motion profiles or poor trajectory planning. A typical type of motion profile is the trapezoidal profile, which consists of an acceleration segment, a constant-speed segment, and a deceleration segment. Theoretically, infinite jerk occurs at the beginning and end of the acceleration and deceleration segments of a trapezoidal profile. This type of profile is often used because industrial controllers used to program machines provide readily available and easy-to-use tools to build simple motion profiles, such as trapezoidal profiles. Meanwhile, the process of building a more complex and efficient profile is more involved, requiring more programming and knowledge. This results in an easy choice for the less-efficient motion profiles. Additionally, some types of trajectories are particularly difficult to accurately follow, e.g., square shapes. This can yield large positioning error even if a well-tuned control scheme is used with a well-designed mechanical system. The difficulty with square shapes resides in accurately tracing the corners, which can result in imperfections to the actual product. This issue is mitigated by trajectory planning, which is a method to calculate the position reference for each axis of a mechanism in such a way that position-following error and jerk are reduced. A trajectory planning approach to mitigate undesirable positioning errors is described in this paper and demonstrated with both simulation and hardware results from the H-Bot robot.
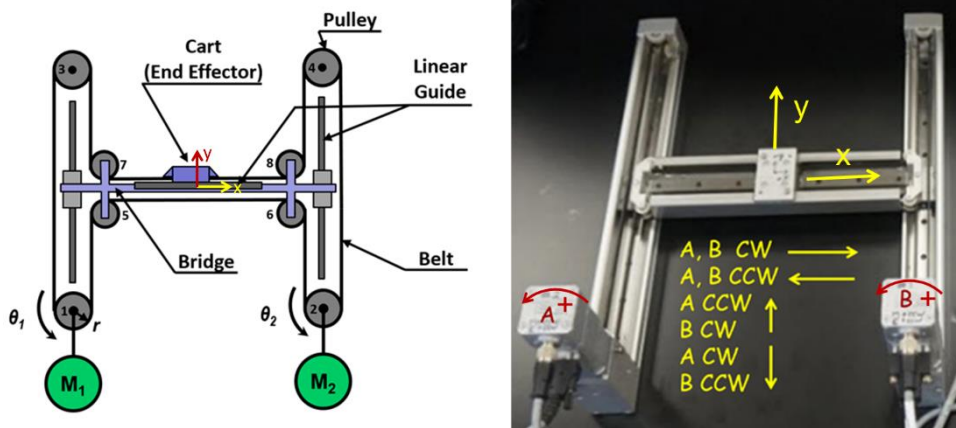

Figure 3. H-Bot Planar Robot

What does optimal performance mean? Suppose the requirement is to move a load from rest to a specified angle θ_f in a specified time t_f. This task can be designed for optimal performance by selecting the commanded motion profile, the transmission connecting the motor and load, and the motor with amplifier. Optimal performance could mean minimum hardware cost, minimum operating cost, minimum energy consumption, minimum operating temperature, etc. Let's consider energy consumption of a brushed dc motor driving a load [9, 12, 13]. The fundamental electrical and mechanical equations of motion are:

$$e = iR + L\frac{di}{dt} + K_e\omega$$

$$J\frac{d\omega}{dt} + B\omega = K_t i - T_d$$

The input voltage is e, i is the armature current, ω is the motor shaft angular velocity, R is the motor resistance, L is the motor inductance, $K_e\omega$ is the speed-dependent voltage called back-emf, $K_t i$ is the electromagnetic torque, J is the motor rotor inertia, B is the motor viscous damping

coefficient, and T_d is the disturbance torque from the load and Coulomb friction. K_e and K_t are the motor back-emf and motor torque constants, respectively. Assume that B is small, as viscous damping is often negligible in practice. The instantaneous power dissipated is i^2R. Solving for i and integrating i^2R from 0 to t_f gives the energy dissipated per cycle, E.

$$J\frac{d\omega}{dt} = K_t i - T_d \qquad i = \frac{J\frac{d\omega}{dt} + T_d}{K_t} \qquad E = \int_0^{t_f} R\, i^2(t)\, dt$$

$$E = \frac{R}{K_t^2}\int_0^{t_f}\left(J\frac{d\omega}{dt}\right)^2 dt + \frac{2R}{K_t^2}\int_0^{t_f} T_d J \frac{d\omega}{dt} dt + \frac{R}{K_t^2}\int_0^{t_f} T_d^2\, dt$$

If the motor starts and stops at rest and T_d is constant, then the only term that depends on the motion profile chosen is:

$$E = \frac{RJ^2}{K_t^2}\int_0^{t_f}\left(\frac{d\omega}{dt}\right)^2 dt$$

What is the optimal velocity profile to minimize motor energy dissipation assuming that T_d is constant? Calculus of variations tells us that the optimum profile is a parabola. The problem then is to find $\omega(t)$ to minimize E subject to several constraints (shown below), but also to be able to trace out shapes with accuracy and without excessive vibration and stress. The solution is a trade-off.

$$E = \frac{RJ^2}{K_t^2}\int_0^{t_f}\left(\frac{d\omega}{dt}\right)^2 dt \qquad \omega(0) = 0 \qquad \omega(t_f) = 0 \qquad \int_0^{t_f}\omega(t)\, dt = \theta_f$$

However, this simple approach allows one to compare different motion profiles and assess energy efficiency. In the case of the H-Bot robot and tracing a square, zero velocity and zero acceleration at the start and end of each square side are essential. The 5th-order polynomial has that capability, while more common profiles, e.g., trapezoidal, do not.

2. TRAJECTORY PLANNING, MOTOR SELECTION, AND CONTROL DESIGN PROCESS

The proposed trajectory planning / motor selection / control design process employs model-based design to improve the process of choosing trajectories and motors for motion applications. Since this method allows selection while in the design stage of the system, trial-and-error tests and physical prototyping are avoided. This reduces the design costs and allows faster delivery of the system.

Each step of this proposed selection process is shown in Figure 4. These steps are described and demonstrated next by creating the trajectory profile, selecting motors, and designing a controller for an H-bot tracing a square shape.
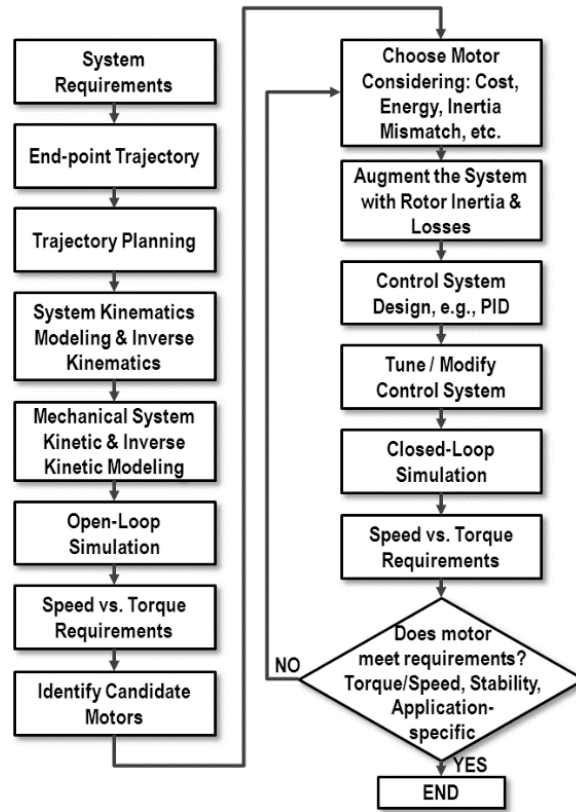
Figure 4. Trajectory Planning, Motor Selection, and Control Design Process

The system requirements define the minimum functionalities and indices of performance that the system needs to achieve. This may include the envelope of motion, trajectory, maximum positioning accuracy, machine cycle time, pay load, and maximum allowable level of vibration and noise. The system requirements are, in general, contained in the functional specifications of the machine. For the H-Bot, the system requirements consist of performing the square shape shown in Figure 5 with diagonal of 0.4 meters in 2 seconds with a maximum position error of 1 mm.
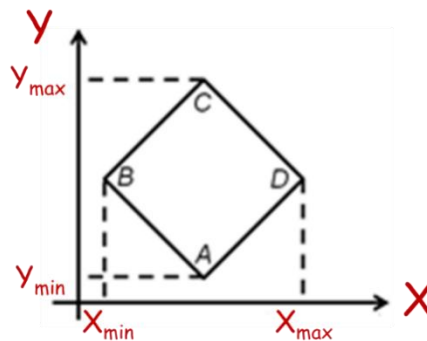

Figure 5. Square Shape Specification

The end-point trajectory is defined here as the motion of the end-effector. Depending on the type of mechanism, the end-point trajectory can be defined in reference to different points on a machine, e.g., at the end-effector for robots, at the platen for presses, and at the load for

conveyors. The end-point or end-effector trajectory is defined for the worst-case trajectory that the end-effector must perform. This worst-case trajectory is identified from the functional specifications of the machine. The worst-case trajectory is, in general, defined as a function of the application requirements and not in terms of the maximum capability of the mechanism. This may allow selecting smaller motor sizes. It is essential to define the end-point trajectory first, because the required torque vs. speed curve used to size the motor depends on it.

If the system needs to perform multiple types of trajectories, designing motors and drives for the worst case is enough. If it is not obvious which trajectory is the worst case, either all, or the most likely worst cases, must be evaluated to identify which one requires the greatest motor and drive utilization. When the mechanism can perform an infinite number of different trajectories, such as a pick-and-place mechanism with a vision system to locate products in random orientation and location, the boundary conditions of the worst-case trajectory need to be determined in order to estimate the worst-case trajectory to be used in the motor selection process.

For this case-study of an H-Bot, the square path with a diagonal of 0.4 meters, traced in 2 seconds, was identified from the functional specification of the machine as the worst-case trajectory for this particular system.

Trajectory planning is the computation of motion profiles for the actuators of automatic machines, e.g., packaging machines, machine tools, assembly machines, metal-forming machines, and industrial robots. Such motion profiles need to be defined in a way to avoid or reduce the amount of mechanical vibration, stress on mechanical and electronic components, electrical and audible noise, stress on motors and actuators, as well as to reduce overshoot response and excessive position error during motion.

The information necessary to compute the trajectory planning is the end-point trajectory and inverse kinematics of the mechanism. Since kinetic models are not necessary at this point, the trajectory planning can be defined at the early stages of the design from sketches with the main dimensions of the moving mechanism. Thus, there is no need for information about masses and mass moments of inertia of the system for trajectory planning.

Trajectory planning is particularly important to reduce position tracking errors in machines such as machine tools while performing certain types of profile. This includes profiles with sharp corners in coordinated-motion systems. A typical example of such a profile is a square shape. The difficulty with square shapes resides in accurately tracing the corners without overshoots or distortions that can cause imperfections to the final product.

A trajectory planning / motor selection / control design approach that can be applied to general types of industrial machines is given next. This approach is demonstrated with an H-Bot tracing a square shape and consists of the following steps.

(1) Identify points in the end-point trajectory at which the velocity changes direction. These are the points in which at least one axis from the same coordinated motion system changes the direction of motion. In the case of the square shape, shown in Figure 5, this occurs at the corners. In corners A and C, the Y axis changes direction, while in corners B and D, the X axis changes direction.

(2) Define the master command.The master command or master reference is used to synchronize the motion of the axes in the system. In industrial applications, this master reference is, in general, a virtual axis, i.e., an axis that only exists in the machine code without any hardware (e.g., drives, motors) associated with it. Its purpose is the

synchronization of the physical axes via a common reference command. The master reference needs to be defined as a profile that helps reduce vibration, mechanical and electrical stresses, and noise on the physical axes. However, the master reference is in general set to constant speed which can yield trapezoidal profiles on the physical axes for certain types of end-point trajectories, and consequently cause these undesirable effects. Thus, the master reference needs to be defined as a motion profile that mitigates these effects. Motion profiles with zero acceleration at the beginning and end of the move yield smoother motion for physical systems. Although various types of profiles could be used to build the master reference, the 5th-order polynomial profile yields a good tradeoff between smoothness and peak velocity. The master reference will consist of segments located between every two consecutive points of the end-effector trajectory that has a change in polarity on the velocity profile. Thus, the master position command (S_M) can be defined for each one of these segments as a 5th-order polynomial profile as follows:

$$S_M = 10\left(\frac{t}{T}\right)^3 - 15\left(\frac{t}{T}\right)^4 + 6\left(\frac{t}{T}\right)^5 \quad [1]$$

T is the desired time to complete each segment and *t* is the instantaneous time. The master command for the H-Bot contains four segments ($\overline{AB}, \overline{BC}, \overline{CD}, \overline{DA}$) located between each corner of the square shape. The master position command has unitary increment from segment to segment. This unitary increment continues until it completes the entire end-point trajectory. Then the master command can either continue the unitary increment into the next machine cycle or be reset back to zero. The duration of each segment is given by *T*. The desired time *T* depends on the time to perform each segment, which may not be the same for all segments.

In the design of a single-axis system, Eq. (1) can be used directly to define the command position of the motor. In case of coordinated systems (i.e., resulting motion depends on two or more axes, such as in a H-Bot), Eq. (1) defines the master command that in used with the geometric equations to calculate the command position of the motors, as described next.

(3) Identify geometric equations. The geometric equations describe the end-point trajectory as coordinates in the Cartesian space. The number of geometric equations is given by the degree of freedom of the end-point trajectory. In the H-Bot case, the end-point trajectory is in the XY plane. Thus, there are two geometric equations, one to describe the motion in X and one in Y. The geometric equations of the H-Bot performing the square shape shown in Figure 5 are as follows:

$$X = \begin{cases} \left[(X_{MAX} - X_{MIN})/2\right](1 - S_M) + X_{MIN} & \overline{AB} \\ \left[(X_{MAX} - X_{MIN})/2\right](-1 + S_M) + X_{MIN} & \overline{BC} \\ \left[(X_{MAX} - X_{MIN})/2\right](-1 + S_M) + X_{MIN} & \overline{CD} \\ \left[(X_{MAX} - X_{MIN})/2\right](5 - S_M) + X_{MIN} & \overline{DA} \end{cases} \quad [2]$$

$$Y = \begin{cases} \left[(Y_{MAX} - Y_{MIN})/2\right]S_M + Y_{MIN} & \overline{AB}, \overline{BC} \\ \left[(Y_{MAX} - Y_{MIN})/2\right](4 - S_M) + Y_{MIN} & \overline{CD}, \overline{DA} \end{cases}$$

X_{MAX}, X_{MIN}, Y_{MAX} and Y_{MIN} are the maximum and minimum displacements in the X and Y directions, respectively. The resulting profiles for X and Y obtained from Eq. (2), with X_{MIN} and

Y_{MIN} equal to zero and the master command (S_M) defined in Eq. (1) for the H-Bot tracing a square shape with diagonal of 0.4 meters in 2 seconds, is shown in Figure 6.
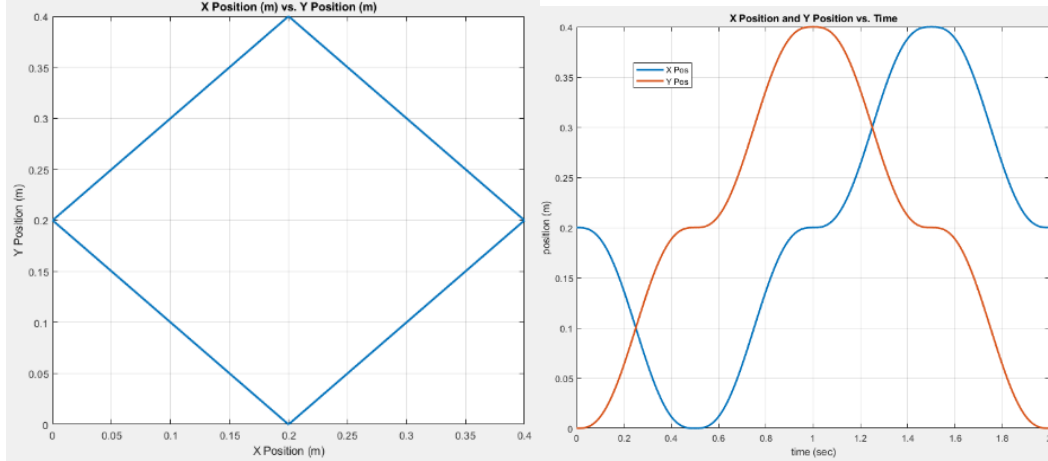

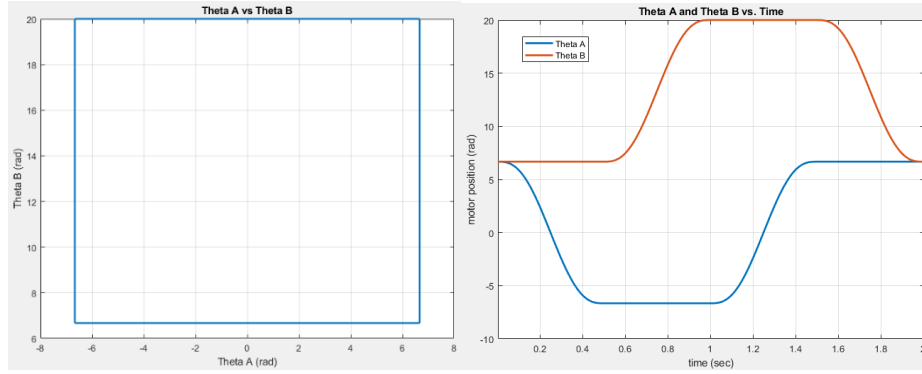Figure 6. X and Y Position Commands for H-Bot Square Shape

The values of *X* and *Y* defined in Eq. (2) from the master command S_M defined in Eq. (1) can now be applied to the inverse kinematics of the system to compute the motor motion profiles.

(4) Inverse Kinematics. The inverse kinematics (geometry of motion) of the mechanism converts the end-point trajectory described in the Cartesian space into the motion profiles that control the motors. In the case of a mechanism with redundant degrees of freedom, e.g., a 7-degree-of-freedom manipulator, infinite possibilities of motion profiles of the motors can be defined to yield the same end-point trajectory. In this case, one of the possible inverse kinematic solutions needs to be chosen to perform the motor selection.

The diagram of an H-Bot is shown in Figure 3 and it works as follows. If one motor stays stationary and the other one rotates, the end effector moves diagonally. If both motors rotate at the same speed in the same direction, the end effector moves left or right. If both motors spin at the same speed in opposite directions, the end effector moves up or down. Accordingly, the inverse kinematics of this type of robot, for an inextensible belt, can be derived and results in Eq. (3).

$$\begin{bmatrix} \theta_A \\ \theta_B \end{bmatrix} = \frac{1}{r}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} \quad \begin{matrix} x,\ y = \text{position of object} \\ r = \text{motor pulley radius} \end{matrix} \quad \begin{bmatrix} x \\ y \end{bmatrix} = \frac{r}{2}\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} \theta_A \\ \theta_B \end{bmatrix}_{[3]}$$

θ_A is the angular position of motor shaft A (ccw+), θ_B is the angular position of motor shaft B (ccw+), X and Y define the position of the end effector in the Cartesian space given by the trajectory planning in Eq. (2), and *r* is the radius of the driving pulley connected to each motor. The resulting profile from the inverse kinematics in Eq. (3) with X and Y computed from the trajectory planning defined in Eq. (1) and (2) is shown in Figure 7.

Figure 7. θ_A and θ_B Motor Position Commands for H-Bot Square Shape

(5) Forward and Inverse Kinetics. The inverse kinetic model (geometry plus all torques and mass moments of inertia) is used to compute the torque that each motor needs to apply to the system to follow the required motor motion profile given by the inverse kinematics. Thus, the input of the inverse kinetic model is the motor angular position profile while the output is the motor torque. The inverse kinetic model is obtained from the kinetic model that can be derived by methods such as Newton-Euler and Lagrange.

The kinetic model of the H-Bot was developed with the following assumptions: horizontally mounted, massless, rigid belt with no slip, rigid bodies, frictionless joints and slides, and inertialessidler-pulleys. By employing the Lagrange Method, the equations of motion are obtained. Lagrange's Equations are as follows:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_i}\right) - \frac{\partial T}{\partial q_i} + \frac{\partial V}{\partial q_i} = Q_i \quad [4]$$

T is kinetic energy, *V* is the potential energy, and Q_i are the generalized forces / torques for each generalized coordinate, q_i. See equation [5]. The generalized coordinates are $q_1 = \theta_A$ and $q_2 = \theta_B$. I_A and I_B are the mass moments of inertia of the driving pulleys, *r* is the radius of each driving pulley, m_1 is the horizontal-moving mass, m_2 is the vertically-moving mass, τ_A and τ_B are the motor torques, and *g* is the acceleration of gravity.

$$q_1 = \theta_A \qquad q_2 = \theta_B$$

$$T = \frac{1}{2}I_A\dot{\theta}_A^2 + \frac{1}{2}I_B\dot{\theta}_B^2 + \frac{1}{2}m_1\dot{x}^2 + \frac{1}{2}m_2\dot{y}^2$$

$$= \frac{1}{2}I_A\dot{\theta}_A^2 + \frac{1}{2}I_B\dot{\theta}_B^2 + \frac{1}{2}m_1\left[\frac{r}{2}\left(\dot{\theta}_A + \dot{\theta}_B\right)\right]^2 + \frac{1}{2}m_2\left[\frac{r}{2}\left(-\dot{\theta}_A + \dot{\theta}_B\right)\right]^2$$

$$V = 0$$

[5]

$$Q_1 = Q_{\theta_A} = \tau_A$$

$$Q_2 = Q_{\theta_B} = \tau_B$$

The Lagrange Equations of Motion are:

$$\begin{bmatrix} \dfrac{r^2}{4}(m_1+m_2)+I_A & \dfrac{r^2}{4}(m_1-m_2) \\ \dfrac{r^2}{4}(m_1-m_2) & \dfrac{r^2}{4}(m_1+m_2)+I_B \end{bmatrix}\begin{bmatrix}\ddot{\theta}_A \\ \ddot{\theta}_B\end{bmatrix} = \begin{bmatrix}\tau_A \\ \tau_B\end{bmatrix}$$

$$\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}\begin{bmatrix}\ddot{\theta}_A \\ \ddot{\theta}_B\end{bmatrix}=\begin{bmatrix}\tau_A \\ \tau_B\end{bmatrix} \qquad \begin{bmatrix} m_{11}s^2 & m_{12}s^2 \\ m_{21}s^2 & m_{22}s^2 \end{bmatrix}\begin{bmatrix}\theta_A \\ \theta_B\end{bmatrix}=\begin{bmatrix}\tau_A \\ \tau_B\end{bmatrix}$$

$$\left[m_{11}-\dfrac{m_{12}m_{21}}{m_{22}}\right]s^2\theta_A = \tau_A - \dfrac{m_{12}}{m_{22}}\tau_B$$

[6]
$$\left[m_{22}-\dfrac{m_{12}m_{21}}{m_{11}}\right]s^2\theta_B = \tau_B - \dfrac{m_{21}}{m_{11}}\tau_A$$

$$I_A = I_B \qquad m_{11}=m_{22} \qquad m_{12}=m_{21}$$

$$\left(Js^2\right)\theta_A = \tau_A - K\tau_B$$

$$\left(Js^2\right)\theta_B = \tau_B - K\tau_A$$

The inverse kinetic model obtained from these equations was implemented in MatLab to obtain the required motor torques. The motor command positions θ_A and θ_B were obtained from the inverse kinematics defined in Eq. (3). For higher fidelity of the results, more complete equations of motion can be derived by including effects such as belt compliance, idler-pulley inertia, and friction. The load mass and any external forces / torques can be added to the equations of motion at this point. The level of complexity included in the model of a mechanism is in general associated with the risk of the design. The risk can be measured in terms of the experience with similar mechanisms, importance to the overall process or machine, amount of innovation in the design, safety concerns, etc. High-risk designs require more complete and accurate models. In this particular case study, the model in Eq. (6) was considered complete enough for purposes of this paper.

(6) Motor Torque vs. Speed Curves. The open-loop simulation is used to estimate the torque vs. speed curves (Figure 8) that will be required from the motors to drive the system through the end-point trajectory. The model for this simulation consists of the motion profile obtained from inverse kinematics and trajectory planning applied to the inverse kinetic model of the system.

If a motor database with the rotor inertia and motor losses is available, these parameters can be added to the inverse kinetic model at this point of the motor selection process, instead of adding these parameters after selecting a motor. Automatic iterations can then be performed with this motor database to better select a motor without significant extra computational requirements. At each iteration loop, the torque vs. speed curve, calculated from the inverse kinetics that now includes the effect of the motor, would be compared to the motor torque vs. speed curve and evaluated.
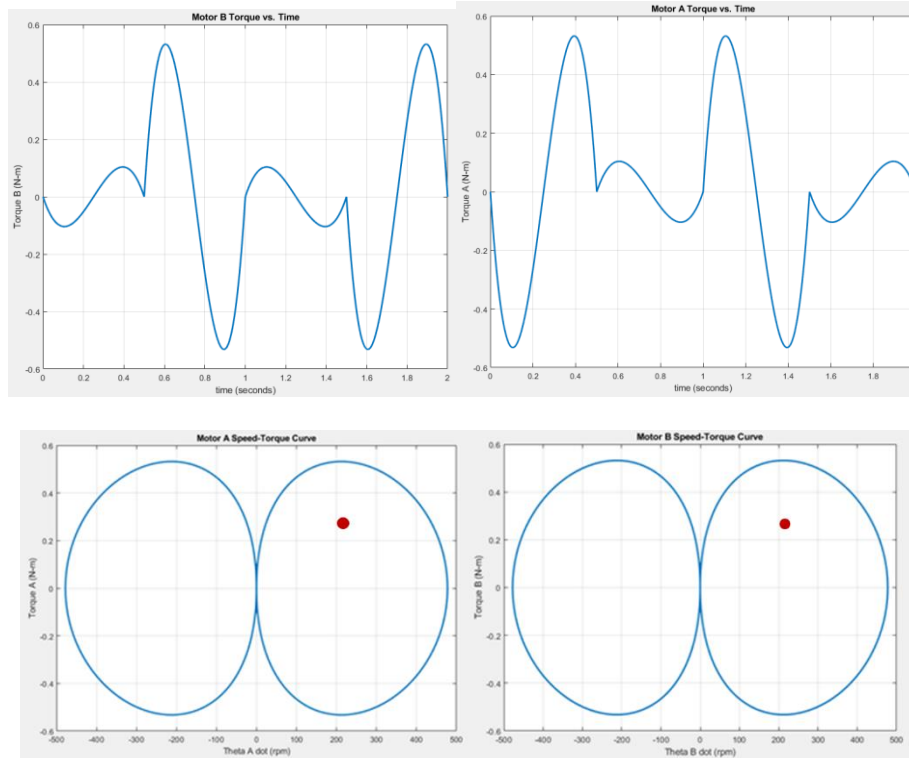
Figure 8. Motors A and B Torque (N-m) vs. Speed (rpm) Curves with RMS Torque and Speed

The open-loop simulation is used to determine the torque vs. speed requirement that each motor must develop for a given motion profile obtained from the worst-case end-point trajectory. The selection of the appropriate motors for the simulated system relies on this estimated torque vs. speed curve. The torque vs. speed curves of the candidate motors must enclose the required torque vs. speed curve of the system. The other condition that needs to be satisfied is that the RMS torque given at the RMS speed of the application must be located below the continuous torque curve of the motor. Otherwise, the motor will overheat and potentially be damaged. The RMS torque (τ_{rms}) and the RMS speed (v_{rms})are calculated as follows:

$$\tau_{rms} = \sqrt{\frac{1}{T} \int_0^T \tau^2(t) dt} \qquad v_{rms} = \sqrt{\frac{1}{T} \int_0^T v^2(t) dt}$$

[7]

$\tau$is the instantaneous torque, v is the instantaneous velocity, and T is the total time of the data sample. The resulting rms torque-speed point is shown by the solid red dot in the speed-torque curves (Figure 8).

(7) Selecting a Motor. The candidate motors (e.g., brushed dc, brushless dc, step) are identified by comparing the required torque vs. speed curve and the rms torque at the rms speed with the torque vs. speed curve obtained from the motor datasheets. The candidate motors are those that enclose the required torque vs. speed curve. The inertia mismatch between the load and motor is, in general, preferred to be kept low (typically less than 10:1) to obtain higher system bandwidth. To keep the inertia ratio low, a gear box or a gearing mechanism may be necessary. Additional requirements, e.g., cost, voltage, energy efficiency, motor size and mounting orientation, will shorten the list of candidate motors.

> The manufacturers of the motors are, in general, limited to those that already supply parts
> to the machine builder.

The type of motor is also part of the motor selection. Torque density, rotor inertia, power, maximum speed, voltage, torque loses, frame type, operating temperature, are some of the characteristics that need to be considered while deciding for a particular motor technology.

The chosen motor becomes an integral part of the system, and its properties, including rotor inertia and torque losses, must be included in the model of the system. Any flexible couplings or gearing contained in the system should also be included in the model. If a gearbox or a transmission mechanism is added to the system to reduce the inertia ratio between the load and motor or increase available torque, not only the gear ratio, but also the losses and inertia of this additional component, must be included in the system.

(8) Control Design Overview. Control is a hidden, enabling technology that is present in almost every engineered system today. Despite this fact, control system design is still mysterious and often falls in the domain of a specialist. Today, every engineer must know how to design, implement, and integrate a control system into a design from the start of the design process. An engineer needs to understand how to balance performance, low cost, robustness, and efficiency to effectively accomplish these goals. Evaluating a design concept is best done through modeling, not by building and testing, as modeling provides true insight on which to base design decisions. There is a hierarchy of models possible of varying complexity and fidelity, but a simple design model which captures essential attributes is the most useful, i.e., dominant dynamics. An integrated control system can enhance a design through stabilization, command following, disturbance and noise rejection, and robustness. All of this can be accomplished through a combined approach, rather than trying to accomplish all with a single feedback controller, as is too often the case. This is shown in Figure 9. The design model is typically used for both feedback and feedforward controller design. However, in practice, the physical system will deviate from that design model.
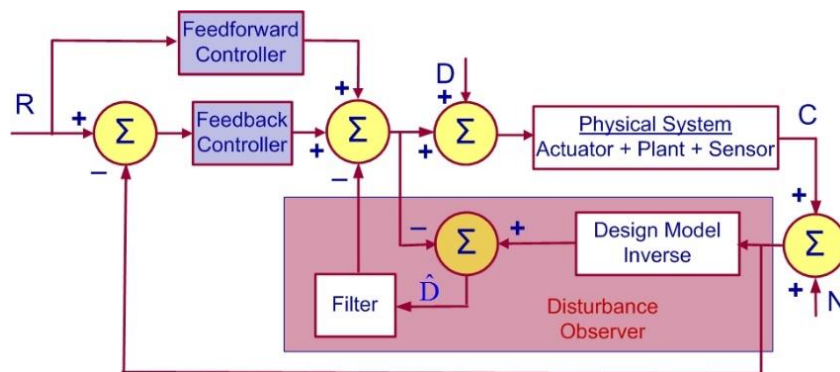

Figure 9. 21st-Century Control Design

A disturbance observer regards any difference between the physical system and the design model as an equivalent disturbance applied to the model. It estimates the disturbance and uses it as a cancellation signal. So, in addition to enhancing disturbance rejection, the disturbance observer makes the physical system behave like the design model over a certain frequency range, thereby simplifying the design of the feedback and feedforward controllers. Since the design model inverse is not realizable, a unity-gain, low-pass filter, specifying the observer bandwidth, is added. Next, the feedback controller is designed solely to force dynamic consistency by mitigating the effects of model uncertainty and disturbances, usually with high gain and integral

control. A common mistake is made in designing the feedback controller for desired output with no regard for robustness, only to find poor performance when applied to the physical system. However, once consistency is enforced, the desired output can be augmented with a feedforward controller, typically the dynamic model inverse, to recover the dynamic delay of the closed-loop system with no effect on stability or properties of the closed-loop system [4].

(9) Control Design Specifics. A typical control scheme for motor control consists of a cascaded control with an inner velocity PI (Proportional-Integral) control and an outer position PI control, as shown in Figure 10. A feedforward velocity loop (FFv) is used to reduce position-following error, while the acceleration feedforward loop (FFa) can be used to reduce velocity-following error. The velocity loop feeds the current loop that controls the energy delivered to the motor to move according to the commanded position profile and feedback response. Two filters are placed between the velocity and current loops to be used as needed: a low pass filter (LPF) to reduce high frequency noise feeding the current loop and a notch filter (NF) to reduce the effect of resonant frequencies. The tuning process must start with the PI gains of the position and velocity loops to make the system robust to load disturbances. Then, the velocity feedforward and acceleration feedforward gains can be used to reduce tracking errors.

For simplicity in simulation and hardware implementation, a PD controller was implemented on each motor of the H-Bot to trace out a square profile. The Simulink block diagram is shown in Figure 11.
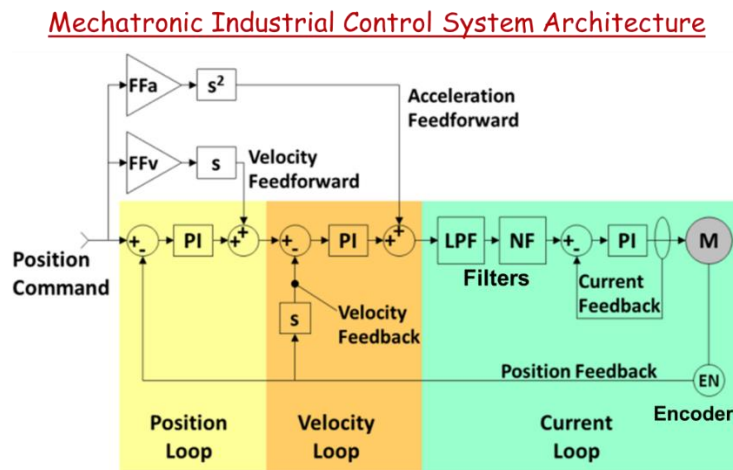

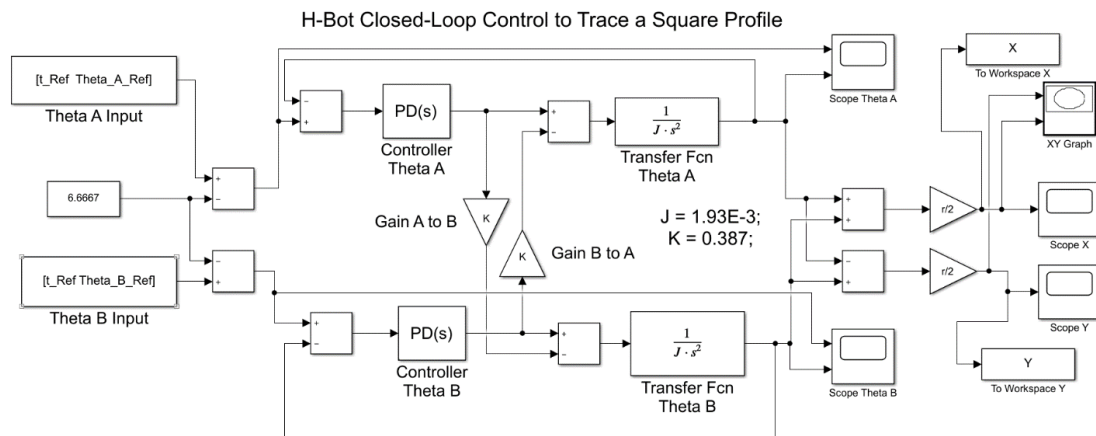Figure 10. Mechatronic Industrial Control System Architecture

H-Bot Closed-Loop Control to Trace a Square Profile

Figure 11. H-Bot PD Feedback Control for Each Motor to Trace a Square Profile

(10) Final Torque vs. Speed Requirements. If the following requirements are met, the selected motor is a valid solution: (a) torque vs. speed curve of the system (including gearbox or transmission, if any) still resides within the torque vs. speed curve of the motor after including the rotor inertia and motor losses and tuning the closed-loop system; (b) the RMS torque and RMS speed are still under the continuous torque curve of the gear-motor; and (3) the system develops a stable response.

If some of these requirements are not met with the selected motor, iterations are needed, i.e., different motors and/or gear boxes need to be evaluated until a valid solution is found. However, in applications where the torque vs. speed requirement is too high, a valid motor/gear box solution may not exist. In this case, the mechanism or the application requirements need to be reevaluated.

3. HARDWARE IMPLEMENTATION WITH ARDUINO / LABVIEWMYRIO

Planar positioning is an important task in industrial applications. The H-Bot uses one long timing belt to transmit the rotation of two stationary motors to end-effector motion. Due to less-moved masses, this system is capable of fast acceleration, and therefore, faster positioning than stacked systems. The use of an elastic transmission element also causes the biggest disadvantage of the system, which is an uncertainty of end-effector position due to stretching in the belt.

In the H-Bot positioning system shown in Figure 12, there are two parallel tracks along which a bridge is lead through on linear ball-bearing blocks. On the bridge there is a third track mounted, which is perpendicular to the first two tracks, on which a cart slides. These tracks form a capitalized H. On each end of the two parallel tracks sits one pulley, where the ones at the lower end are directly attached to the motor shaft. On each end of the track on the bridge, there are also two pulleys. An open timing belt is guided around these eight pulleys. The open ends are both attached to the cart, which runs on the bridge. The system forms a parallel drive configuration, meaning that the actuator drive system is not an open kinematic chain. This parallel drive setup enables the rotational motion of the two stationary motors to transform into a linear *y*-motion of the bridge and a linear *x*-motion of the cart relative to the moving bridge. The overlapping of these two linear motions creates the *XY*-motion of the end effector. Step motors were chosen for this application because of availability.
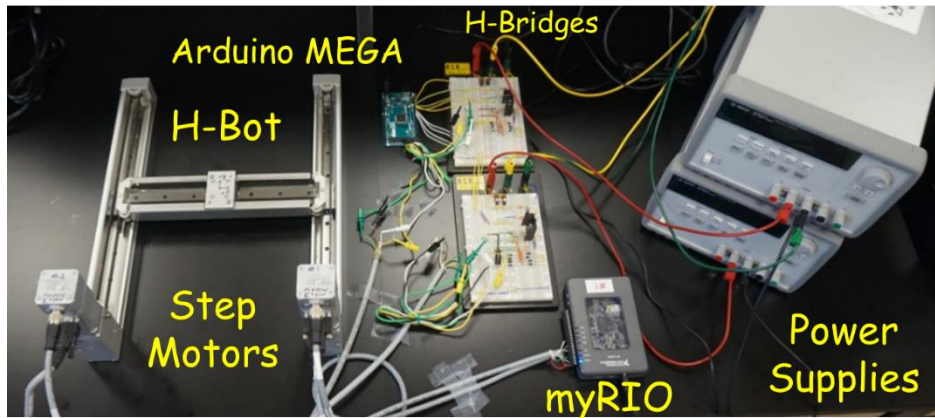
Motor		
Nominal voltage	[V DC]	48
Nominal current	[A]	1.8
Max. rotational speed[1)]	[rpm]	1,740
Holding torque	[Nm]	0.5
Stepper angle	[°]	1.8 ±5%
Winding resistance	[Ω]	1.75 ±10%
Winding inductance	[mH]	5.4
Output mass moment of inertia	[kgcm^2]	0.082
Radial load on shaft	[N]	20
Axial load on shaft	[N]	7
Mass moment of inertia of rotor	[kgcm^2]	0.082

Figure 12. H-Bot Hardware SystemFesto Hybrid Bipolar Step Motor with 500 cpr Encoder

Shown in Figure 13 is the real-time control Simulink diagram for X and Y motion profiles of the end effector which is auto-code-generated for the Arduino. Also shown is the circuit diagram for the Arduino / L298 dual full-bridge driver chip. The National Instruments MyRIOwas used to easily decode the encoder signals and generate the required plots.
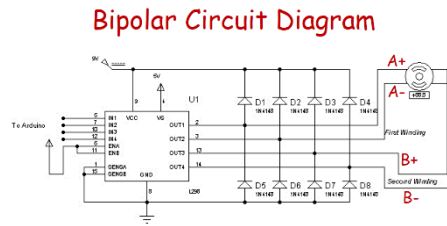
Bipolar Circuit Diagram
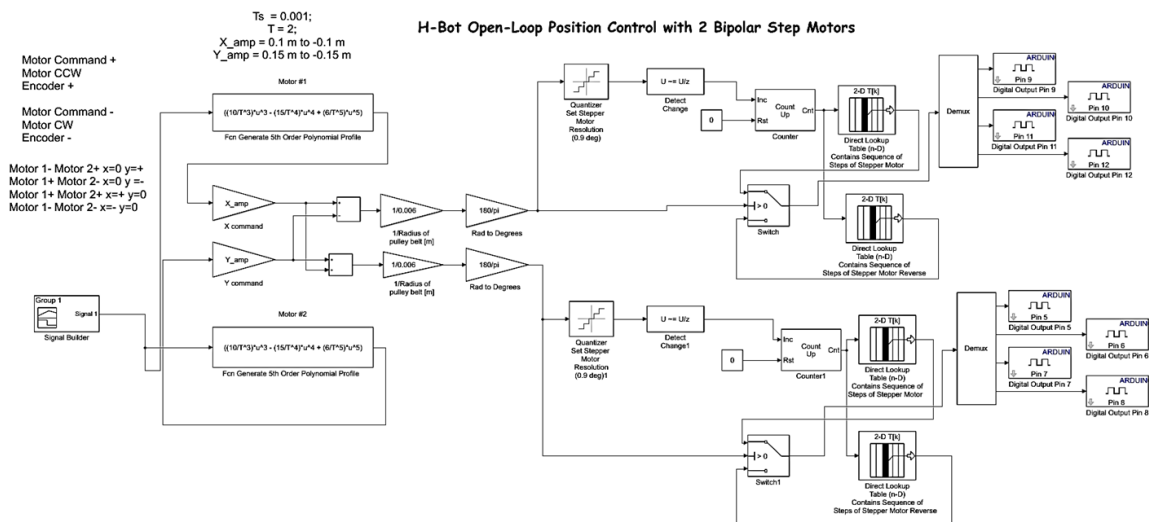
Figure 13. Real-Time Simulink Control Diagram and Bipolar Step Motor Circuit Diagram

4. CONCLUSION

Accurate motion control in industrial machines heavily depends on trajectory planning and the appropriate selection of the motors controlling the axes of the machine. A model-based design approach was presented for (1) trajectory planning that leads to accurate positioning and (2) optimized selection of motors prior to building a prototype. As planar positioning is an important task in industrial applications, a two-axis, single-belt-driven, H-frame planar positioning robot called an H-Bot was built. The proposed approach was demonstrated via modeling, analysis, control-design simulation using MatLab / Simulink, and hardware implementation using the Arduino and LabVIEW MyRIO to follow a demanding square profile. As shown below, the choice of the 5th-order polynomial motion profile, together with selection of bipolar step motors, met the desired accuracy with smooth torques delivered by the motors.
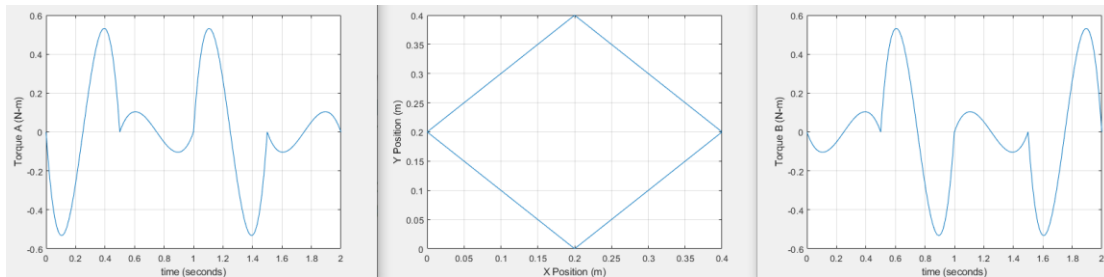
REFERENCES

[1]   K. Craig, "Modeling and Simulation for Motor Selection," Design News, June 2011.
[2]   K. Craig, "Mechatronics System Design Process," Design News, July 2011.
[3]   K. Craig, "Modeling: Essential Key to Mechatronic Design," Design News, January 2007.
[4]   K. Craig, "Control Design: Pervasive and Perplexing," Design News, October 2010.
[5]   K. Craig, "Mechatronify Common Mechanisms," Design News, August 2012.
[6]   K. Craig, "So You Want To Build An H-Bot," Design News, May 2011.
[7]   K. Craig, "Trajectory Planning with Electronic Cams," Design News, November 2010.
[8]   A. M. d. Silva, P. A. Voglewede, and K. C. Craig, "Integrated Trajectory Planning, System Modeling, and Control Design for Optimized Motor Selection," presentation at the ASME 5th Annual Dynamic Systems and Control Conference, Fort Lauderdale, FL, 2012.
[9]   Y. Wang, Y. Zhao, S. Bortoff, and K. Ueda, "A Real-Time Energy-Optimal Trajectory Generation Method for a Servomotor System," IEEE Transactions on Industrial Electronics, Vol. 62, No. 2, February 2015.
[10] R. Isermann, "Modeling and Design Methodology for Mechatronic Systems," IEEE/ASME Transactions on Mechatronics, vol. 1, p. 13, March, 1996 .
[11] C. W. d. Silva, Mechatronics: A Foundation Course. New York: CRC Press, 2010.
[12] William Palm, Modeling, Analysis, and Control of Dynamic Systems, 2nd Edition, Wiley, 1999.

[13]   Aderiano DaSilva, "Mechatronics Design Process with Energy Optimization for Industrial Machines," Ph.D. Dissertation, May 2015, K. Craig Advisor, Marquette University.

[14]   A. AbdelHamid, M. Abdeldayem, and M. Mabrouk, "Low-Cost X-Y Positioning System Using Stepper Motor," 18th International Conference on Applied Mechanics and Mechanical Engineering, April, 2018, Vol. 18.

[15]   K.S. Sollmann, M.K. Jouaneh, and D. Lavender, "Dynamic Modeling of a Two-Axis, Parallel, H-Frame-Type XY Positioning System," IEEE/ASME Transactions on Mechatronics, Vol. 15, No. 2, pp. 280-290, 2010.

AUTHOR

Kevin Craig attended the United States Military Academy at West Point, NY, earned varsity letters in football and baseball, and graduated with a B.S. degree and a commission as an officer in the U.S. Army. After serving in the military, he attended Columbia University and received the M.S., M.Phil., and Ph.D. degrees. While in graduate school, he worked in the mechanical-nuclear design department of Ebasco Services, Inc., a major engineering firm in NYC, and taught and received tenure at both the U.S. Merchant Marine Academy and Hofstra University. While at Hofstra, he worked as a research engineer at the U.S. Army Armament Research, Development, and Engineering Center (ARDEC) Automation and Robotics Laboratory. He received the 1987 ASEE New Engineering Educator Excellence Award, a national honor. In 1989, he joined the faculty at Rensselaer Polytechnic Institute (RPI). At RPI, he served as Director of Core Engineering, the first two years of the School of Engineering, and as Chair of the Engineering Science Interdisciplinary Department. As a tenured full professor of mechanical engineering, he taught and performed research in the areas of mechatronic system design and the modeling, analysis, and control of multidisciplinary engineering systems. With significant continuous funding from both industry and government, he developed the Mechatronics Program at RPI, which included an extensive teaching and research laboratory and several undergraduate and graduate courses in mechatronics. He collaborated extensively with the Xerox Mechanical Engineering Sciences Laboratory (MESL), an offshoot of Xerox PARC, during this time. During his 18 years at RPI, he graduated 37 M.S. students and 20 Ph.D. students. While at RPI, he authored over 30 refereed journal articles and over 50 refereed conference papers. Emphasis in all his teaching and research was on human-centered, model-based design, with a balance between theory and best industry practice. At RPI, he received the two highest awards conferred for teaching: the 2006 School of Engineering Education Excellence Award and the 2006 Trustees' Outstanding Teacher Award. From 2007 to 2014, he wrote a monthly column on mechatronics for practicing engineers in Design News magazine. Over the past 30 years, he has conducted hands-on, integrated, customized, mechatronics workshops for practicing engineers nationally and internationally, e.g., at Xerox, Procter & Gamble, Rockwell Automation, Johnson Controls, Brady Corp., Pitney Bowes, and Siemens Health Care in the U.S., and at Fiat and Tetra Pak in Italy. He is a Fellow of the ASME and a member of the IEEE and the ASEE. After a national search, in January 2008, he was chosen to be the Robert C. Greenheck Chair in Engineering Design, a $5 million endowed chair, at Marquette University.   His mission was to integrate multidisciplinary design and discovery learning throughout the entire college, in all years and in all departments. He transformed students, faculty, curricula, and facilities throughout the college and created a new engineering education mindset and culture for innovation. He was given the 2013 ASEE North-Midwest Best Teacher Award and the 2014 ASME Outstanding Design Educator Award, a society award. He graduated his 21st Ph.D. In the fall of 2014, he returned to the Hofstra University School of Engineering and Applied Science as a tenured full professor of mechanical engineering. He is the Director of the Mechatronics Laboratory, which he created with $1M funding from NYS / Hofstra, and also the Director of the Center for Innovation, a new center he created to collaborate with business and industry to foster innovation, where all intellectual property (IP) belongs to the sponsor. Recent clients include ThermoLift, P&G, and Toro. He has taught graduate courses as an Adjunct ME Professor at both Stony Brook University and NYU Tandon School of Engineering.