Matthew Tran
EE149
MT1

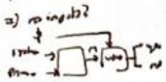
- Ch 2 - Continuous Dynamics
  - Newton's Law - $F(t) = M\ddot{x}(t)$
    - technique $x(t) = x(0) + \int_0^t \dot{x}(\tau)d\tau$
  - Torque - $T(t) = \frac{d}{dt}(I(t)\dot{\theta}(t))$
    - element for each axis/pair
  - model-order reduction - simplify model, reduce dimensions
  - continuous time system
    - box w/ input and output == actor
    - cascade composition - series
    - can have multiple input/output
  - causal - current and past inputs
  - strictly causal - output doesn't depend on present
  - memoryless - only dep on current
  - linear - superposition
  - time invariant - don't care about delay
  - LTI - ↑ ↑
  - BIBO stable - bounded input ⇒ bounded output
  - feedback - self name
- Ch 3 - Discrete Dynamics
  - events occur at instants
  - pure signal - ∈ {present, absent}
  - instantaneous reactions, event triggered
  - type - possible values for port
  - valuation - assign value to port
  - state - the condition, controls reactions
  - state space - possible values?
  - Finite State Machine (FSM)
    - finite number of states
    - initial state
    - transition
      - arrow between states
      - labeled guard/action
      - not mentioned = implicit absent
      - boolean terms - ¬, ∨, ∧
      - output: name := value
      - output valuation: name = value
      - stutter - all absent, no change
    - default transition - true / ... transition func
    - five tuple (States, Inputs, Outputs, update, initialState)
      $(s(n+1), y(n)) = update(s(n), x(n))$
    - Mealy - action on transition, we use this
    - Moore - action in state
    - deterministic - 1 transition possible per input
    - receptive - ≥ 1 transition on each input
      - default transition implicit == always receptive
  - Extended FSM
    - add variables (init value on init transition)
      guard/output action ... new format
      set action(s) ← after top part
    - allows compact notation
    - reachable states
  - Nondeterminism
    - apparently we allow a possible path to never terminate
    - legit a probability could be 0 that's allowed
    - stochastic model - add probability
    - Five tuple - updates → possibleUpdates
  - Execution trace
    s → s → .... graphical better
  - Computation Tree
    like ↑ but a tree
- Ch 4 - Hybrid Systems
  - outputs/actions between transitions, not any more
  - inputs/outputs present anytime
  - state refinement - like FSM inside FSM
  - Timed Automata
    - has clock modeled as $\dot{s}(t) = 1$, $s(t)$ is clock value
    - mode = state
  - Higher order Dynamics
    - ex. sticky masses, bouncy ball
  - Supervisory Control - mode transitions
    - low level control - time based inputs
    - forms a control system
- Ch 5 - Composition of State Machines
  - Concurrent Composition
    - side-by-side - no communication } equivalent
    - synchronous - react at same time }
    - asynchronous - nondeterministic, may miss inputs
      - semantics 1 (interleaving semantics) - either react, not both
      - semantics 2 - A, B, or both react
      - can add scheduling so that it doesn't miss inputs
    - shared variables - many issues ... nondeterministic
      - synchronous interleaving semantics - arbitrary order reaction
      - add priority?
    - cascade/serial composition - same ordering
  - Hierarchical FSM
    - innermost reacts first ... pred
    - preemptive transition - ignore inner ones
    - reset transition - always at initial state
    - history transition - remembers last state

- Ch 7 - Sensors and Actuators
  - linear - $f(x) = ax$
  - affine - $f(x) = ax + b$
  - range - L, H, saturated
  - dynamic range
    - precision p - smallest measurable change
      $D = \frac{H-L}{p}$ $D_{dB} = 20\log_{10}\left(\frac{H-L}{p}\right)$
  - quantization - n-bits, $D_{dB} = (20\log_{10}2)n \approx 6n$ dB
  - oversampling - higher sample rate, LPF, reduce noise
  - noise - $x'(t) = x(t) + n(t)$
    accounted imperfection and quantization = noise
    Noise RMS $N = \lim_{T\to\infty}\sqrt{\frac{1}{2T}\int_{-T}^{T}(n(\tau))^2 d\tau}$
    $SNR_{dB} = 20\log_{10}\left(\frac{X}{N}\right)$ // X ... ?
  - sampling - usually fixed frequency
    - aliasing!
    - Nyquist-Shannon sampling theorem
  - harmonic distortion - sensitivity depends on magnitude
    2nd harmonic $f(x) = ax + b + d_2x^2$
  - Signal conditioning
    - stuff
  - Common Sensors
    - accelerometer - proper accel = observer in free fall ... ≠ gravity
    - anemometer
    - GPS
    - WiFi fingerprinting ... etc.
    - Bluetooth
    - gyroscope
    - mic
- Ch 9 - Memory Architecture
  - volatile - need power
  - SRAM - faster, power, lose
  - DRAM - cheaper, has refresh time
  - EEPROM
  - flash - slower than RAM
    - NOR - low density, fast read
    - NAND - more durable, block read
  - virtual memory
  - memory map
    - everything is a "memory access"
    - bit banding - use address for 1 bit, faster
  - register - few but fast
  - scratchpad - more data, close to fast
  - cache - duplicate of data
  - MMU - illusion of lots of memory
  - cache - valid, tag, block
    - tag, set index, block offset
    - direct mapped - conflict misses
    - set associative - each set multiple lines
      - LRU, FIFO
    - fully associative - 1 set, expensive
  - little or big endian
  - stack
    - stack pointer
    - stack frame - data for func
  - MPU - protection
  - heap, garbage collection
  - static vs not always related in C
    C in practice, ...
- Ch 10 - Input and Output
  - Pulse width modulation
  - GPIO - careful current, noise
    - Schmitt trigger
  - UART
  - USB
  - JTAG, boundary scan, SWD
  - I^2C, SPI, PCIe, FireWire, HDMI
  - DMA
  - Interrupts
    - HW, SW, exception
    - calls ISR, vector table?,
    - volatile - tells compiler to always fetch
  - Atomicity
    - simultaneous access bad
  - device drivers - use interrupts, hard

  - Extra
    - affine inputs don't preserve superposition
    - FSMs consider ALL CASES for always receptive
    - extended FSM - state also include variables
    - (uint32_t *)

Matthew Tran
EE149
MT2

- Ch 11 Multitasking
  - multitasking - simultaneous exec of tasks
  - imperative language - sequence of ops (ex. C)
  - threads - share memory
  - cooperative multitasking - task triggers scheduler
  - timer lead to store state, so have clock w/ jiffy time to ISR
  - may be interrupted between 2 atomic ops - race condition
    - use mutex (mutual exclusion locks), semaphore
    - one per time critical section
    - careful of deadlock (one order is one fix)
  - memory consistency
    - sequential consistency
      not guaranteed by compiler (or even HW)
  - insidious error - hard to test, hard to detect
  - process - unique memory space
    - MMU handles translation
    - IPC - files, msg passing (shared memory)
  - careful unbounded queues
- Ch 12 Scheduling
  - real time - ordering, timing, deadline constraints
    - also processor assignment for multicore
  - design vs run time - when to decide?
  - fully static scheduler - purely design time
  - static order scheduler - assign order - design time; timing - run time
    - aka off-line scheduler
  - static assignment scheduler - assignment - design time; order + timing - runtime
    - runtime scheduler picks who to run next } online schedulers
  - fully-dynamic scheduler - all runtime
  - preemption - interrupt running task
    - can do when blocked
  - task model - assumptions (i.e. task termination)
  - arrival of tasks - task arrive after start
  - sporadic - repetition + unpredictable timing
  - precedence constraints; preconditions
  - release time r_i (arrival time)
  - start time s_i (exec start)
  - finish time f_i
  - response time $o_i = f_i - r_i$
  - execution time e_i (time running)
    - usually assume WCET (worst case exec time)
  - deadline d_i (hard or soft)
  - fixed vs. dynamic priority
  - feasible schedule - $f_i < d_i$
  - utilization - time not idle < 0 feasible
  - max lateness $L_{max} = \max(f_i - d_i)$
  - makespan (total time) $= M = \max f_i - \min r_i$
  - Rate Monotonic (RM) scheduling
    - optimal w/ respect to feasibility for fixed priority unicore
    - higher priority to smaller period
    - assume negligible context switch
    - utilization bound $\mu \le n(2^{1/n} - 1) \Rightarrow$ feasible RM
  - Earliest Deadline First (EDF) aka Horn's alg
    - earliest due date (EDD) aka Jackson's alg no arrival
      - optimal w/ minimize max lateness
    - dynamic priority scheduling
      - optimal w/ feasibility among?
      - minimize max lateness, fewer preemptions
    - up to 100% util
    - EDF w/ precedences - use graph
    - (Latest Deadline First (LDF) - no arrival of tasks, but optimal feasibility } same?
    - (EDF* - modified deadlines $d_i' = \min(d_i, \min_{j \in D}(d_j' - e_j))$, $j \in D$ = dep on i
  - priority inversion - where, use inheritance
  - priority ceiling - acquire if higher priority than all the ceiling
  - Hu level scheduling - priority based on greatest sum of exec time of path
    - critical path - explore paths that use time
  - Richard's anomalies - nonmonotonic, increase, dec time ⇒ longer!
    - very common in practice
- Ch 13 Invariants and Temporal Logic
  - invariant always true
  - Linear Temporal Logic (LTL)
    - property on traces, paths
    - propositional logic formula
      $x \Rightarrow y \equiv \neg x \vee y$
      $\equiv \neg y \Rightarrow \neg x$ contrapositive
    - trace $q_0, q_1, \ldots$
    - ϕ - holds for q_0
      $G\phi$ - ϕ holds for all traces $j \ge 0$
      $F\phi$ - ϕ holds for some suffix $j \ge 0$
      $X\phi$ - ϕ holds for next trace $q_1, q_2, \ldots$
      $\phi_1 U \phi_2$ - ϕ_1 holds until ϕ_2 true
        ϕ_1 may stop after
- Ch 14 Equivalence and Refinement
  - type refinement (B of A)
    - can replace A w/ B
    1. fewer inputs ≤ } ports
    2. more outputs ≥
    3. more input values ≥ } types
    4. fewer output values ≤
  - language refinement (B of A)
    - set of all behavior L
    - $L(B) \subseteq L(A)$ == A more
    - LTL for A holds for B
  - simulation refinement (M_2 of M_1)
    == sim abstraction M_1 of M_2
    - aka M_1 simulates M_2
    - matching, M_2 picks, M_1 matches ←
    - simulation relation $S \subseteq S_1 \times S_2$ pairs in each step
    - transitive, non-unique
    - $L(M_2) \subseteq L(M_1)$
  - bisimulation
    - not same as simulate both direction
    - game either can move
- Ch 15 Reachability Analysis and Model Checking
  - open vs closed system ⇒ no inputs?
  - formal verification



    - error trace - counterexample
  - Verify $G p$?
    - finite state ⇒ not easy bc many states
    - DFS? need compression
    - symbolic abstractions - use prop logic, set ops
    - localization reduction - hide vars to simplify
    - CEGAR - hide all vars, then add on what needed
  - safety vs. liveness prop methods
  - Automata
- Ch 16 Quantitative Analysis
  - WCET vs BCET
  - types vs. loose bound
  - threshold property - bound?
  - average case analysis
  - basic block - like atomic code
  - Control Flow Graph (CFG)
    - directed
    - call & return edges to funcs (dashed)
    - can help find WCET
  - Flow control - paradox?
  - ILP vs. arbitrate integers
- Ch 17 Security and Privacy
  - confidential - being secret
  - integrity - unmodified
  - authenticity - know who
  - availability - is there
  - DOGS, AES, RSA
  - encryption == unknown
  - dig sig
  - message authentication code - symmetric
  - Diffie-Hellman = asymmetric
  - PTP