

Abstrakt

V mém projektu jsem se pokusil navrhnout a realizovat řídicí systém pro hobby CNC frézku. Můj systém odstraňuje dva hlavní nedostatky současných systémů – omezené možnosti generování výstupních signálů a nevhodné zacházení se strojem z pohledu dynamiky pohybu.

Můj systém je postaven na osobním počítači a STM32F4 Discovery kitu.

Hypotéza

Omezené možnosti generování signálů jsou dány použitím stolního počítače, který pro tyto účely není navržen. Řešením tohoto problému je přidání specializovaného hardwaru. V mém případě STM32F4Discovery kitu, který obsahuje ARM mikrokontrolér. Tento kit bude obstarávat veškeré činnosti náročné na přesné časování. Počítač bude sloužit pouze jako uživatelské rozhraní a bude řešit výpočetně náročné úkony.

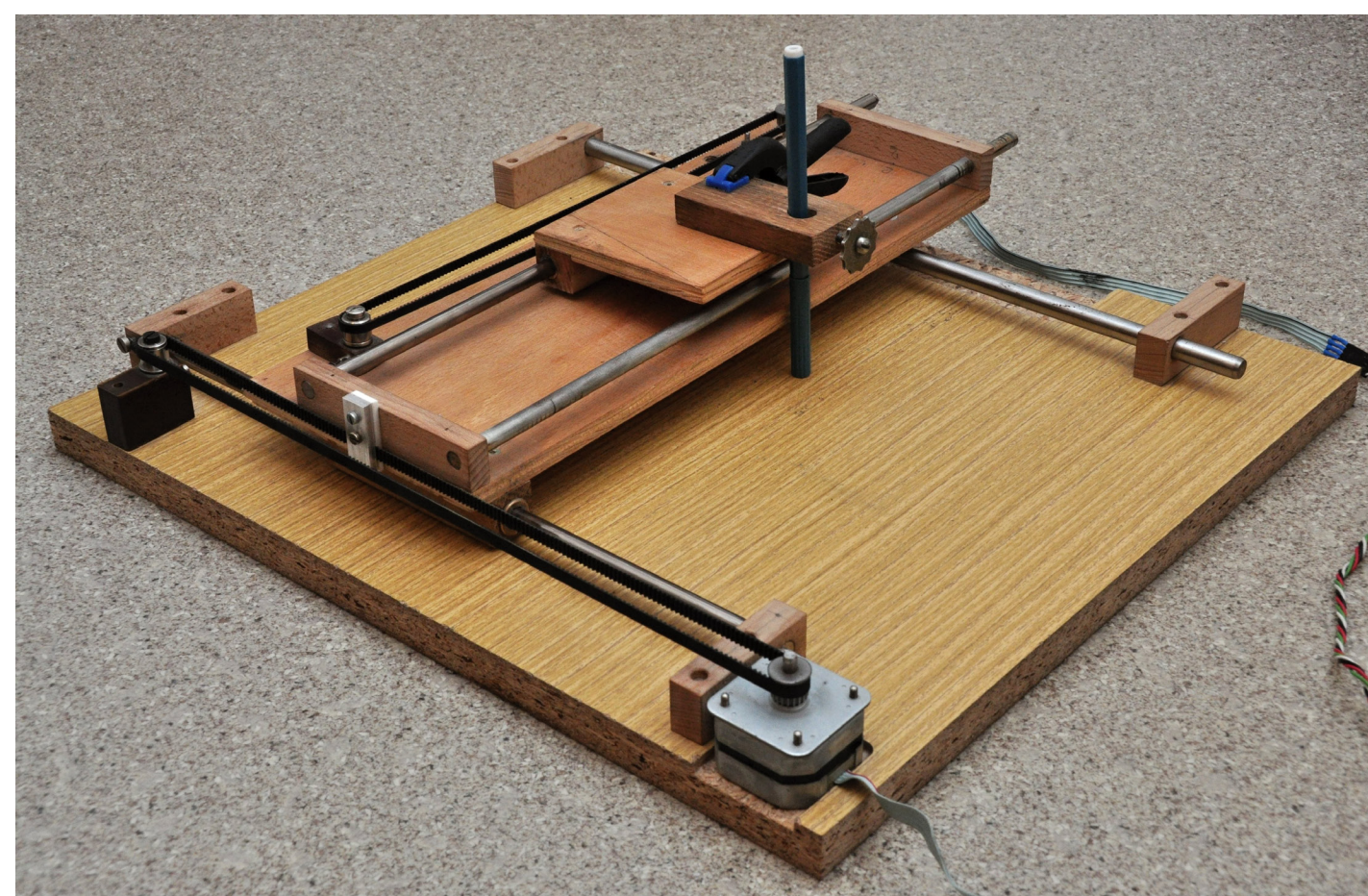
Abych odstranil namáhání stroje, musím analyzovat pohyb stroje z pohledu dynamiky, zjistit příčiny a navrhnout řešení. Cílem je odvodit fyzikální model, který bude charakterizovat pohyb vhodný pro stroj z pohledu dynamiky.

Materiály

Jelikož je daná problematika relativně hodně specializovaná a neustále se vyvíjející, nelze k ní nalézt mnoho materiálů.

Velmi hodnotné podněty jsem získal na sdružení českých amatérských stavitelů CNC strojů c-n-c.cz. Informace z tohoto fóra nelze považovat vždy za pravdivé, ale poskytly mi široký přehled do problematiky.

Hodně cenných informací o požadovaném chování řídicího systému jsem získal z manuálů k profesionálním systémům – Heidenheim TNC410, Siemens Sinumerik 810T. Dále jsem se inspiroval chováním nejpokročilejšího řídicího systému pro hobby CNC stroje – Linuxem CNC.



Obrázek 2: Testovací plotter

Fyzikální model

Na začátku mých úvah jsem analyzoval fyzikální model většiny současných řídicích systémů a snažil se najít jeho nedostatky.

Podstata tohoto modelu je zobrazena na grafu 1. Tento model využívá pro rozjezd a brzdění stroje lineární akcelerační křivku (závislost rychlosti na čase). Aby bylo této závislosti dosaženo, je nutné působit konstantním zrychlením. Zde nastává úskalí tohoto modelu. Z Newtonových pohybových zákonů vyplývá, že pro udělení zrychlení je nutné působit silou, která je přímo úměrná zrychlení. Zrychlení zde prudce vzrůstá a prudce klesá – tím pádem i působící síla. Vzniká ráz v pohybu. Navíc pohony stroje nejsou schopny okamžitě začít působit plnou silou.

Lineární pohyb

Pro odvození modelu lineárního pohybu jsem si stanovil, že zrychlení má průběh:

$$a = A \sin \frac{\pi t}{T}$$

Postupným integrováním jsem získal vztahy pro okamžitou rychlost a okamžitou dráhu:

$$v = \frac{AT}{\pi} \left(1 - \cos \frac{\pi t}{T} \right) + v_0 \quad s = \frac{AT}{\pi^2} \left(\pi t - T \sin \frac{\pi t}{T} \right)$$

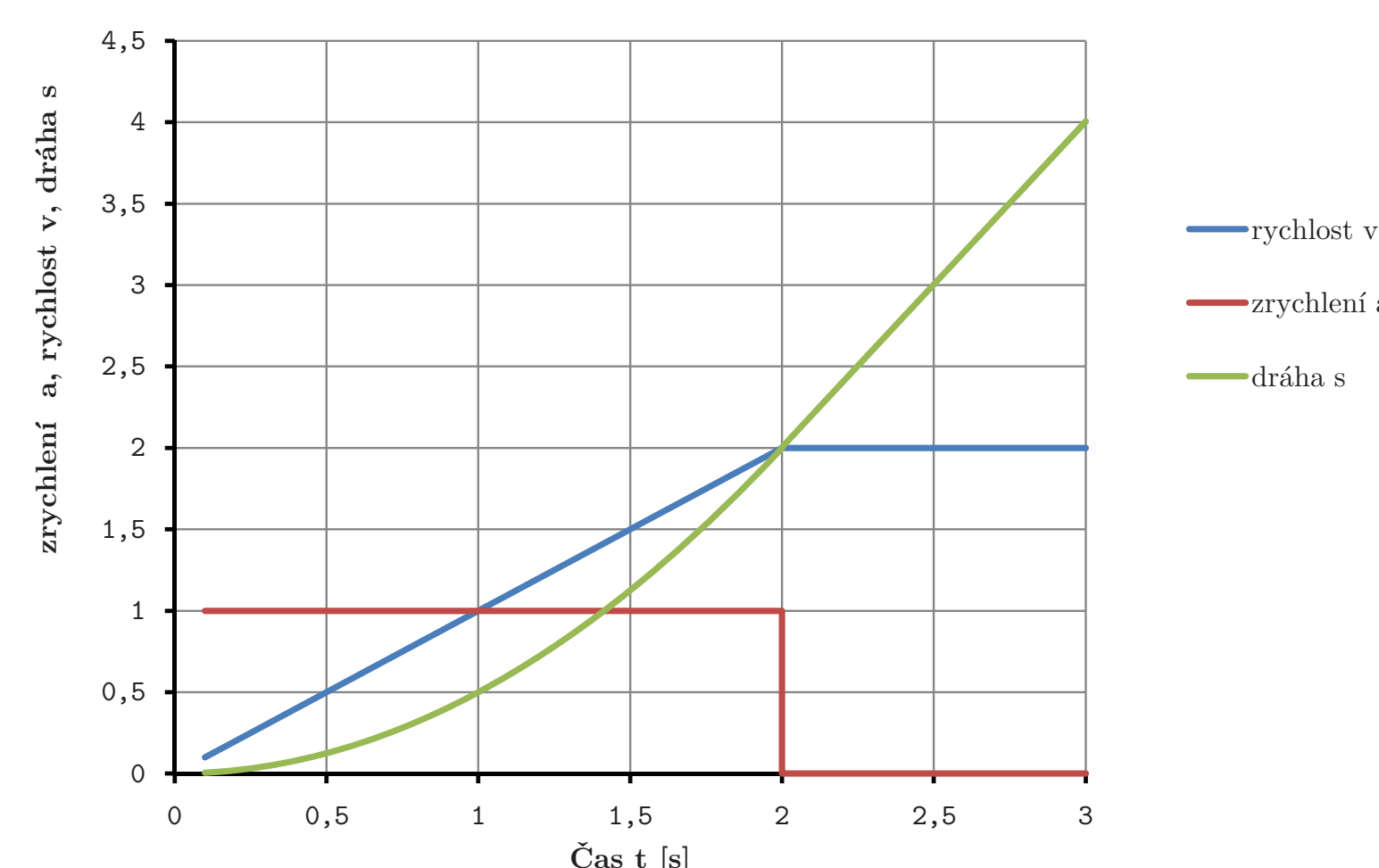
Z těchto vztahů jsem si odvodil vztah pro dobu pohybu, pokud je pohyb omezen zrychlením:

$$T = \frac{\pi (V - v_0)}{2A}$$

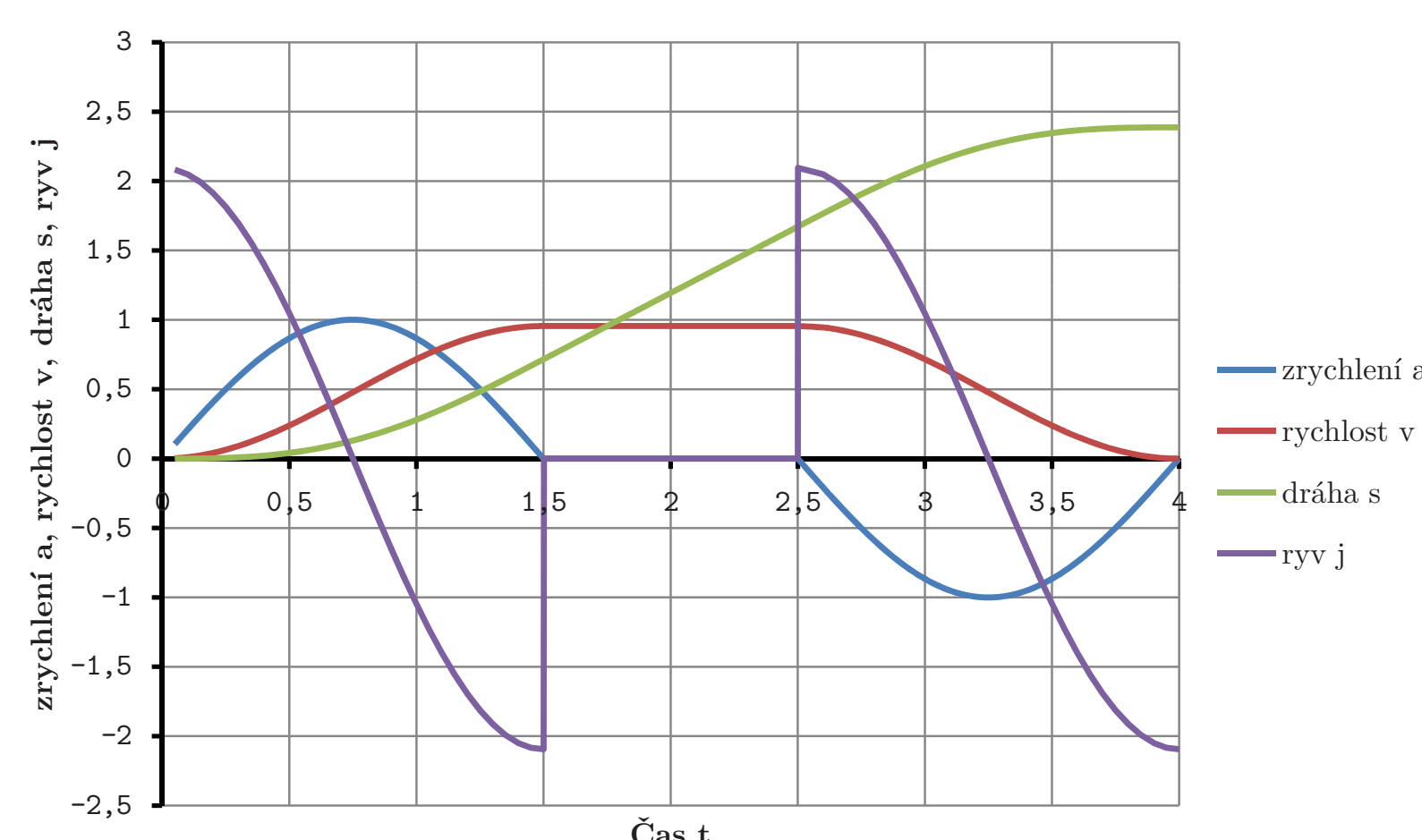
Následně jsem derivací mého zrychlení dostal vztah pro průběh ryvu v pohybu. Poté jsem určil jeho maximum a jeho vhodným dosazením jsem získal vztah pro dobu pohybu při omezení ryvem.

$$T = \frac{\pi}{2} \sqrt{\frac{2(V - v_0)}{J}}$$

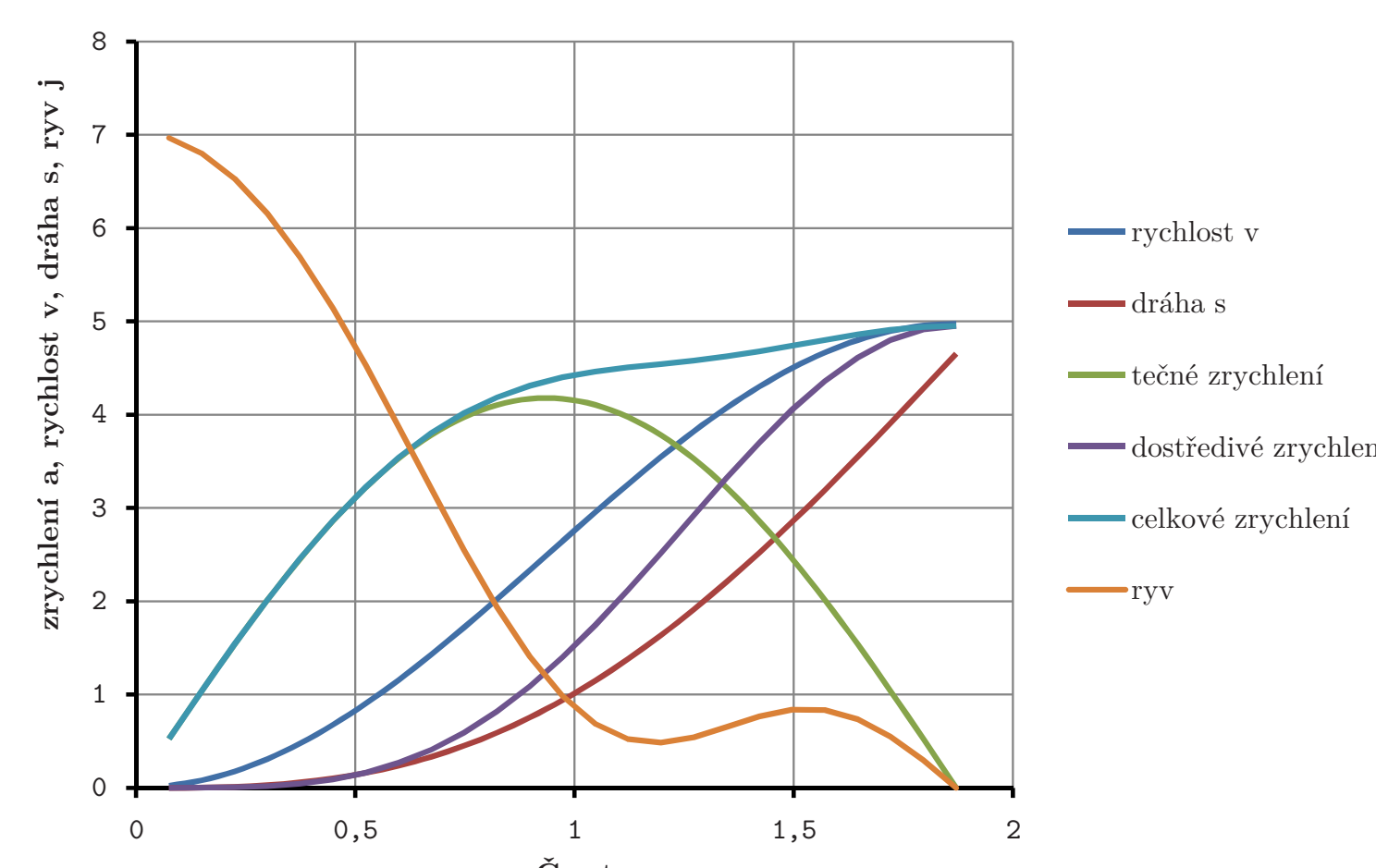
Ilustrační průběh pohybu lze vidět na grafu 2.



Graf 1: Klasická akcelerační křivka



Graf 2: Akcelerační S-křivka pro lineární pohyb



Graf 3: První možný případ akcelerační křivky na pohybu po oblouku

Můj model tedy vychází z takzvaných S-křivek – jedná se o akcelerační rampu, která je mírně prohnutá do písmene „S“ (viz graf pro lineární pohyb). Pro můj systém jsem si stanovil akcelerační rampu ve tvaru funkce $y = 1 - \cos x$. Abych této závislosti docíll, musí mít zrychlení sinový průběh – získávám tím tedy postupný nárůst a pokles působící síly.

Omezujícími faktory pro můj systém jsou maximální přípustná rychlost (omezení konstrukcí stroje), maximální špičkové zrychlení (omezení maximální působící síly) a maximální špičkový ryv (charakterizuje maximální změnu zrychlení, resp. působící síly).

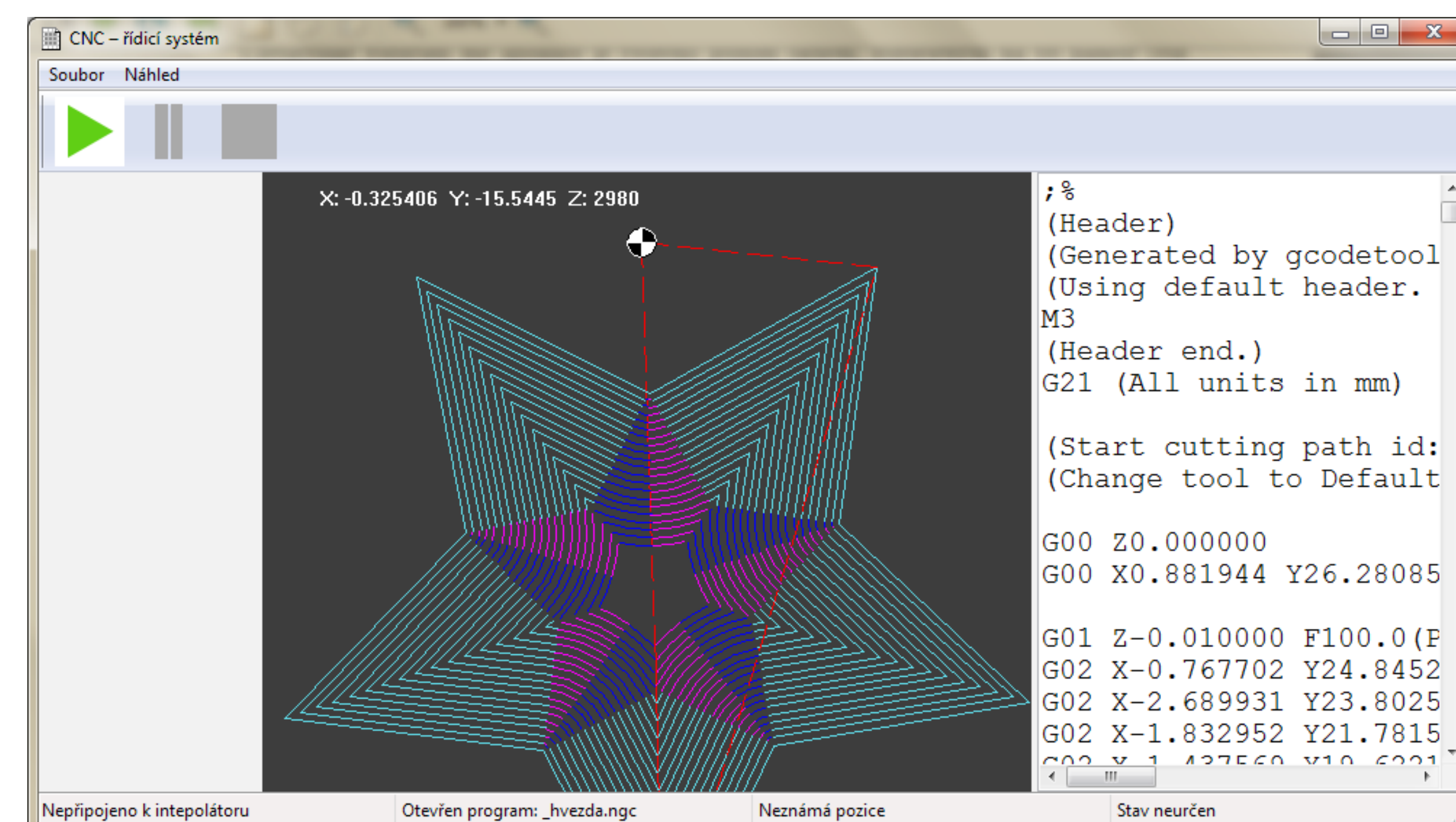
Pohyb po oblouku

Při pohybu na oblouku musím uvažovat i vznikající dostředivé zrychlení. Jelikož dostředivé zrychlení roste se vzrůstající rychlostí, nelze zachovat jeho sinový průběh – zrychlení neodezní. Mým cílem je zachovat sinový průběh zrychlení pro tečné zrychlení a tím zachovat i plynulý nárůst a plynule odeznění celkového zrychlení.

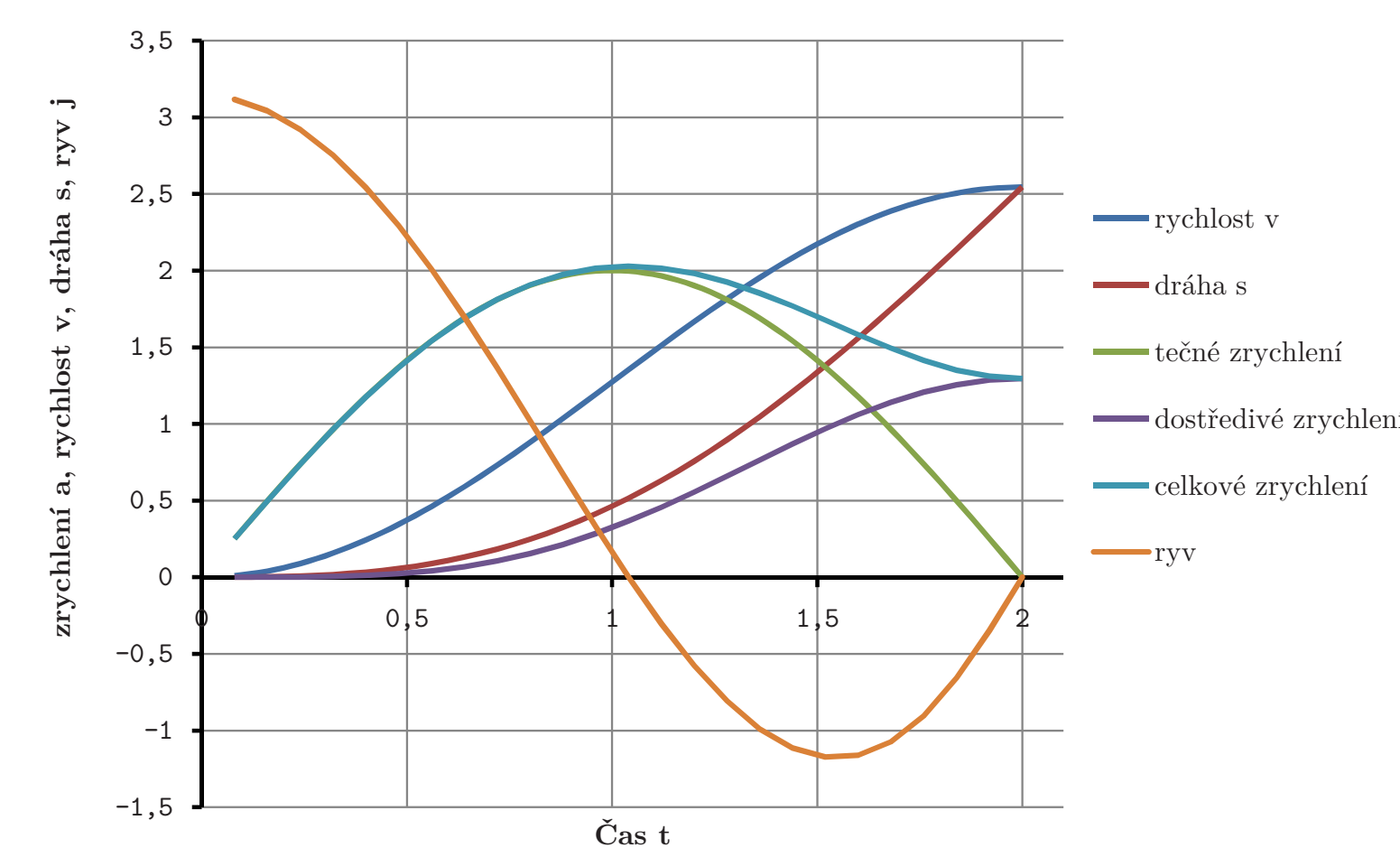
$$\mathbf{a_k} = \mathbf{a} + \mathbf{a_d} \implies |a_k| = \sqrt{|a|^2 + |a_d|^2}$$

Na základě pozorování jednotlivých situací jsem dospěl k závěru, že existují dva možné případy průběhu (grafy 2 a 3) – v prvním je maximální zrychlení přibližně v čase $t = 0$, v druhém dosahuje maxima v čase $t = T$. Následně může ryv dosahovat svého maxima v čase $t = \frac{T}{2}$ nebo přibližně $t = \frac{3T}{4}$.

Pro tyto jednotlivé situace jsem si odvodil příslušné vztahy, z nichž však některé nemají obecné řešení a je třeba je řešit numericky.



Obrázek 1: Screenshot uživatelského rozhraní



Graf 3: Druhý možný případ akcelerační křivky na pohybu po oblouku

Co je to ryv?

Ryv je fyzikální veličina popisující změnu zrychlení. Její jednotkou je 1 m s^{-3} . Je definována jako:

$$\mathbf{j} = \frac{d\mathbf{a}}{dt} = \frac{d^2\mathbf{v}}{dt^2} = \frac{d^3\mathbf{d}}{dt^3}$$

Realizace

Systém se skládá ze dvou částí – aplikace pro počítač zajišťující uživatelský vstup a výpočet rychlosti – a nezávislého hardwarového interpolátoru postaveného na STM32F4Discovery kitu, který provádí samotnou interpolaci a generuje řídicí signály. Počítač s interpolátorem komunikuje pomocí USB.

Aplikace pro počítač bere za vstup G-kód, čímž umožňuje snadné napojení na CAM programy. Z G-kódu podporuje všechny nejpoužívanější funkce včetně veškerých funkcí pro korekci nástroje. Uživateli zobrazuje grafický náhled drah.

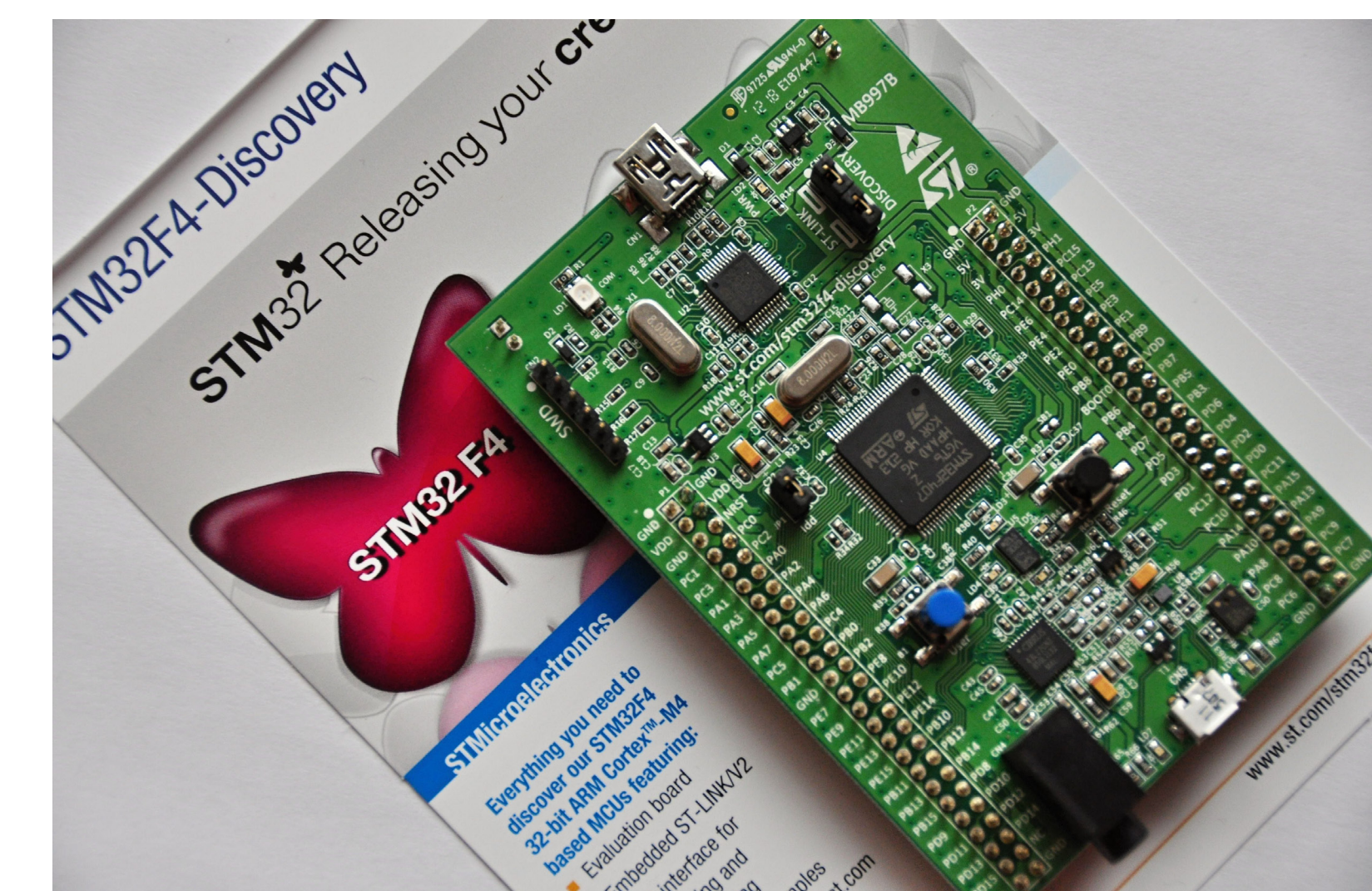
Aplikace pro počítač byla napsána v jazyce C++11 s použitím knihoven WxWidgets a WinUSB.

Interpolátor je postaven na výše uvedeném kitu, který disponuje ARM mikrokontrolérem. Po připojení k počítači se hlásí jako vendor-specified zařízení. Interpolátor provádí veškeré výpočty v kombinované aritmetice (jak fixed, tak floating-point). Jejich kombinací dosahuje požadovaného kompromisu mezi přesností a rychlostí. Pro pohony generuje step-dir signál.

Závěr

Můj systém úspěšně řeší oba problémy současných řídicích systémů. Není však vhodný pro nasazení do praxe, protože mu chybí spousta drobné funkcionality a robustní uživatelské rozhraní. Nachází se spíše ve fázi technického demo, které ukazuje možnosti řízení CNC strojů z pohledu dynamiky.

Řízení z pohledu dynamiky v mém systému v některých ohledech předčích nejpokročilejší systém pro hobby stroje – Linux CNC (též EMC). Tento systém je open-source a poskytuje robustní základ. Zvažuji, zda-li by nestálo za to zkusit můj fyzikální model zakomponovat do tohoto systému.



Obrázek 3: STM32F4Discovery kit